



دانشکده مهندسی کامپیوتر
دانشگاه صنعتی شریف

تمرینات Take Home امتحان نهایی
Take Home Assignments of Final Exam of

سیستم های فازی

Fuzzy Systems

استاد: دکتر سعید باقری و دکتر لیلا شریف
Prof.: Dr. Saeed Bagheri & Dr. Leila Sharif

وحید مواجی 83205947
Vahid Mavaji 83205947

تیر 1384
July 2005

فهرست مطالب

1.....	سوال اول
8.....	سوال دوم
12.....	سوال سوم
19.....	سوال چهارم
21.....	سوال پنجم
28.....	سوال ششم
30.....	سوال هفتم
42.....	سوال هشتم
46.....	سوال نهم
49.....	سوال دهم
62.....	سوال یازدهم
66.....	سوال دوازدهم
68.....	سوال سیزدهم
72.....	سوال چهاردهم

1. روشهای استنتاج فازی را که می شناسید بنویسید و برای هر کدام یک مثال ارائه دهید.

دو روش برای نتیجه گیری از روی یک مجموعه قاعده وجود دارد: استنتاج مبتنی بر ترکیب قواعد و استنتاج مبتنی بر قواعد جداگانه. در استنتاج مبتنی بر ترکیب قواعد، تمامی قواعد موجود در پایگاه قواعد فازی در یک رابطه فازی $U \times V$ ترکیب شده و آنگاه بدیده یک قاعده اگر – آنگاه فازی تنها نگریسته می شود. در ابتدا می بایست بفهمیم آنچه که یک مجموعه از قواعد می گوید چیست و آنگاه از عملگرهای منطقی مناسب برای ترکیب آنها استفاده کنیم.

دو نظر مخالف در رابطه با آنچه که یک مجموعه از قواعد فازی می گوید وجود دارد. اولین نظر در این باره، قواعد را بدیده عبارتهای شرطی مستقل نگاه می کند. اگر ما این نقطه نظر را بپذیریم، آنگاه عملگر معقول و مناسب برای ترکیب قواعد، اجتماع می باشد. دومین نظر در این باره، قواعد را بدیده ترکیبهای شرطی که به شدت بهم وابسته بوده، نگاه کرده به نحوی که شرایط تمامی قواعد بدین خاطر که مجموعه قواعد یک حقیقت را می گویند می بایست ارضا گردند. اگر ما این نقطه نظر را بپذیریم، آنگاه باید از عملگر اشتراک برای ترکیب قواعد استفاده کنیم. نظر دوم گرچه ممکن است عجیب به نظر برسد ولی برای بعضی استلزام ها نظیر استلزام گودل دارای معنا می باشند.

فرض کنید $Ru^{(l)}$ یک رابطه فازی در $U \times V$ و نشان دهنده قاعده اگر – آنگاه بدین صورت: $Ru^{(l)} = A_1^l \times \dots \times A_n^l \rightarrow B^l$ می دانیم که $A_1^l \times \dots \times A_n^l$ یک رابطه فازی در $U = U_1 \times \dots \times U_n$ می باشد که بصورت زیر تعریف شده است:

$$\mu A_1^l \times \dots \times A_n^l(x_1, \dots, x_n) = \mu A_1^l(x_1) * \dots * \mu A_n^l(x_n)$$

که $*$ یک عملگر t – نرم است. اگر اولین نظر را در مجموعه قواعد بپذیریم آنگاه M قاعده رابطه، بعنوان یک رابطه فازی Q_M در $U \times V$ تفسیر شده که بدین ترتیب تعریف می شود:

$$Q_M = \bigcup_{l=1}^n Ru^{(l)}$$

این ترکیب، ترکیب ممدانی نامیده می شود. اگر از سمل + برای نمایش s – نرم استفاده کنیم آنگاه می توان نوشت:

$$\mu_{Q_M}(x, y) = \mu_{Ru^{(1)}}(x, y) + \dots + \mu_{Ru^{(M)}}(x, y)$$

برای دومین نقطه نظر درباره مجموعه قواعد، M قاعده اگر – آنگاه فازی بعنوان یک رابطه فازی Q_G در $U \times V$ تفسیر شده که بدین شکل تعریف می گردد:

$$Q_G = \bigcap_{l=1}^M Ru^{(l)}$$

و یا

$$\mu_{Q_G}(x, y) = \mu_{Ru^{(1)}}(x, y) * \dots * \mu_{Ru^{(M)}}(x, y)$$

که * نشان دهنده t – نرم می باشد. این ترکیب، ترکیب گودل نامیده می شود. فرض کنید A' یک مجموعه فازی دلخواه در U و ورودی موتور استنتاج فازی باشد. آنگاه با در نظر گرفتن Q_M یا Q_G بعنوان یک قاعده اگر – آنگاه فازی و با استفاده از مودس پوننس تعمیم یافته، خروجی موتور استنتاج فازی را اگر از ترکیب ممدانی استفاده کنیم، بدین ترتیب بدست می آوریم:

$$\mu_{B'}(y) = \sup t[\mu_{A'}(y), \mu_{Q_M}(x, y)], x \in U$$

و اگر از ترکیب گودل استفاده کنیم:

$$\mu_{B'}(y) = \sup t[\mu_{A'}(y), \mu_{Q_G}(x, y)], x \in U$$

انتخاب های متعددی برای موتور استنتاج فازی وجود دارد:

1. استنتاج مبتنی بر ترکیب و استنتاج مبتنی بر قواعد جداگانه و در میان استنتاج مبتنی بر ترکیب، استنتاج ممدانی یا گودل.
2. استلزام دینس – رشر، استلزام لوکاشیویکز، استلزام زاده، استلزام گودل یا استلزام های ممدانی.
3. عملگرهای مختلفی برای t – نرم ها و s – نرم ها.

یک مدل برای استنتاج هم مدل ساگنو می باشد. در مدل ساگنو، فرم قواعد پایگاه معرفت با مدل ممدانی فرق می کند. در این مدل تلاش بر مدلسازی مهارت شخص خبره بوده و قواعد به فرم کلی زیر می باشند:

if x is A_1 then y is $f_1(x)$

if x is A_2 then y is $f_2(x)$

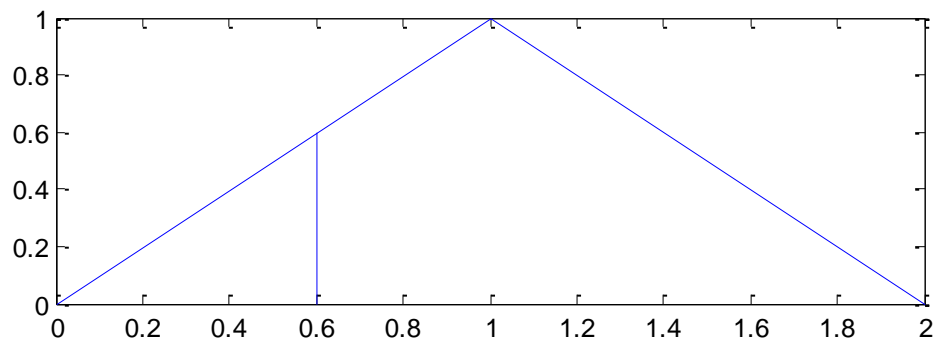
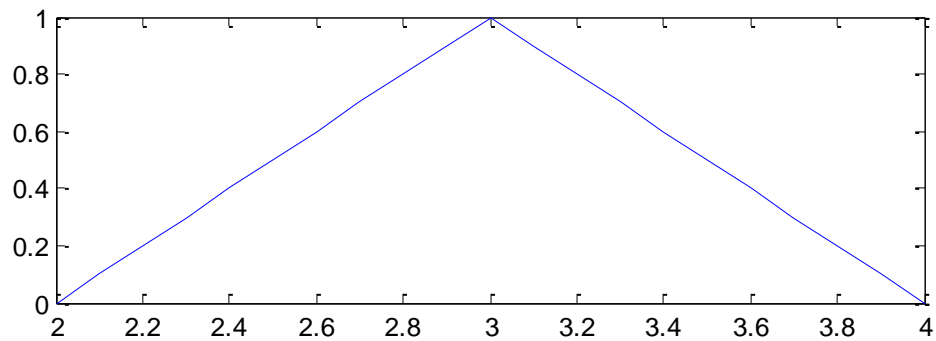
که f تابعی بر حسب متغیرهای ورودی کنترل کننده است. در این حالت نیز ممکن است به ازای برخی حالت‌های ورودی، دو قاعده قابل استفاده باشند و باید خروجی y از ترکیب دو قاعده ای بدست آید که قسمت شرط آن قاعده ها توسط ورودی کنترل کننده برآورده شده است.

در استنتاج شرطی اگر $\mu_R(x, y)$ تابع عضویت یک رابطه شرطی روی $X \times Y$ باشد و $\mu_{\tilde{A}}(x)$ ، تابع عضویت مشاهده صورت گرفته، روی X باشد، براساس قانون GMP تابع عضویت نتیجه استنتاج، $\mu_{\tilde{B}}(y)$ برابر است با:

$$S_n[T_n(\mu_R(x, y), \mu_{\tilde{A}}(x))]$$

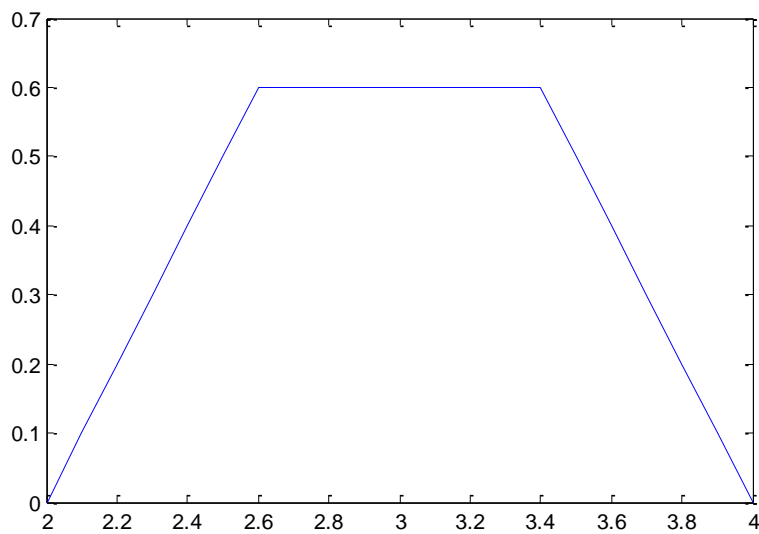
$$\forall x \in X, y \in Y$$

که در این عبارت S_n و T_n به ترتیب همان S -نرمها و T -نرمها می باشند. حال برای شکل 1-1 که شامل یک مجموعه فازی و یک مشاهده تک فازی به درجه اعتقاد 0.6 است، نتیجه استنتاج را برای حالات مختلف S_n و T_n به دست می آوریم.



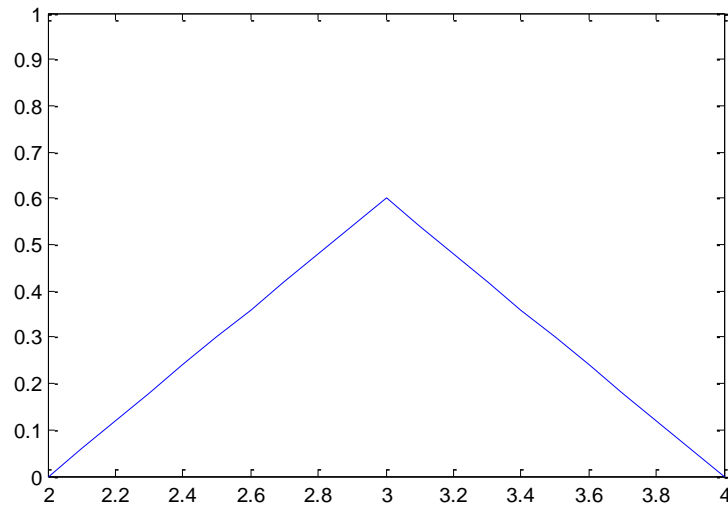
شکل 1-1: تابع عضویت رابطه فازی (بالا) و مشاهده یگانه مربوط به آن (پایین)

$\max(\bar{A})$ به جای S_n و \min به جای T_n (استنتاج max-min): شکل 2-1 نتیجه استنتاج را نشان می دهد:



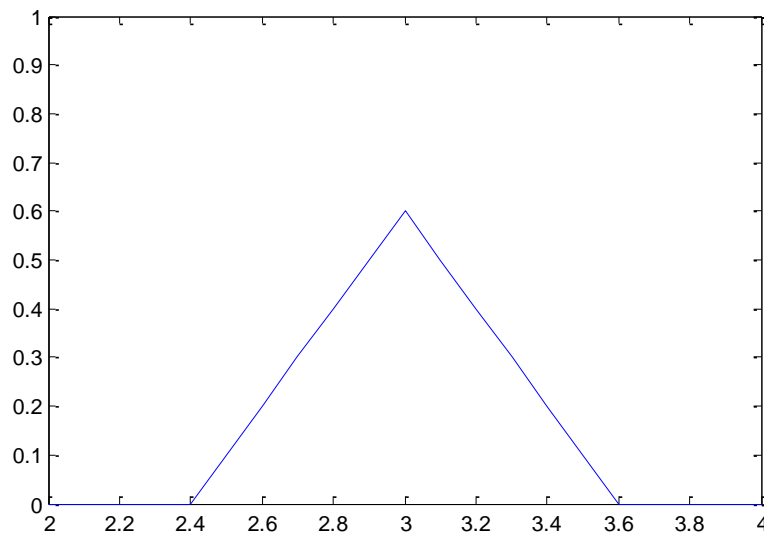
شکل 2-1 : استنتاج max-min

ب) \max به جای S_n و ضرب به جای T_n (استنتاج max-dot): شکل 3-1 نتیجه استنتاج را نشان می دهد:



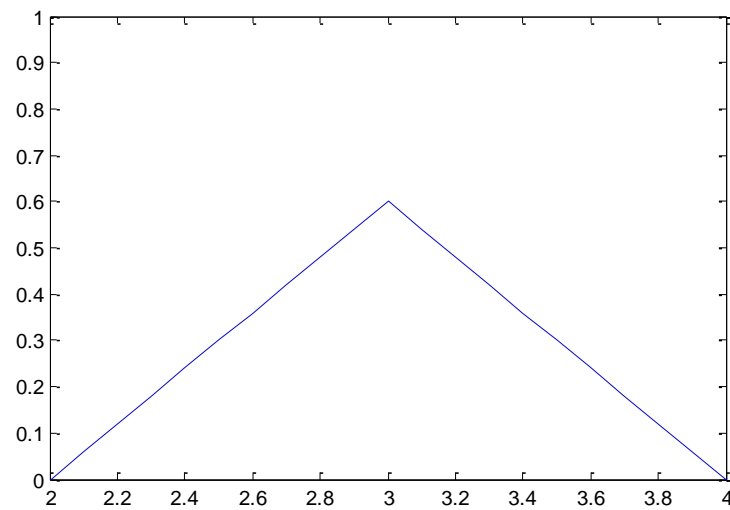
شکل 3-1: استنتاج max-dot

پ) \max به جای S_n و bounded difference به جای T_n : شکل 4-1 نتیجه استنتاج را نشان می دهد:



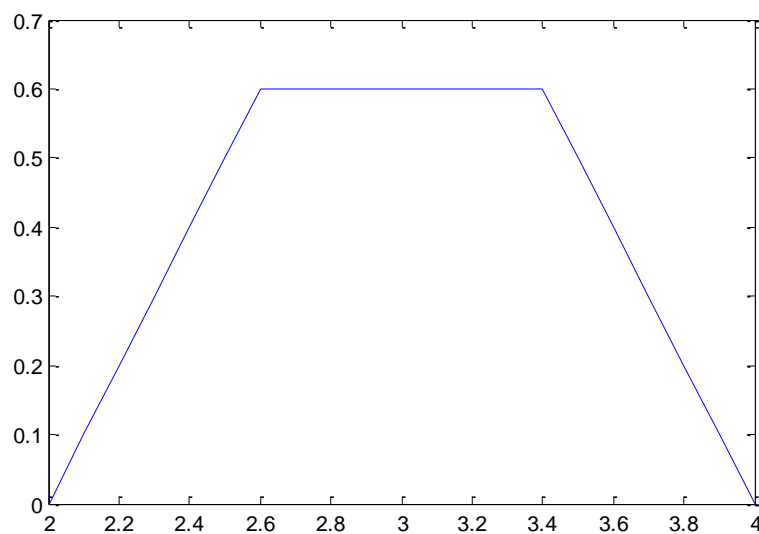
شکل 4-1: استنتاج max-bounded difference

ت) bounded sum به جای S_n و ضرب به جای T_n : شکل 5-1 نتیجه استنتاج را نشان می دهد:



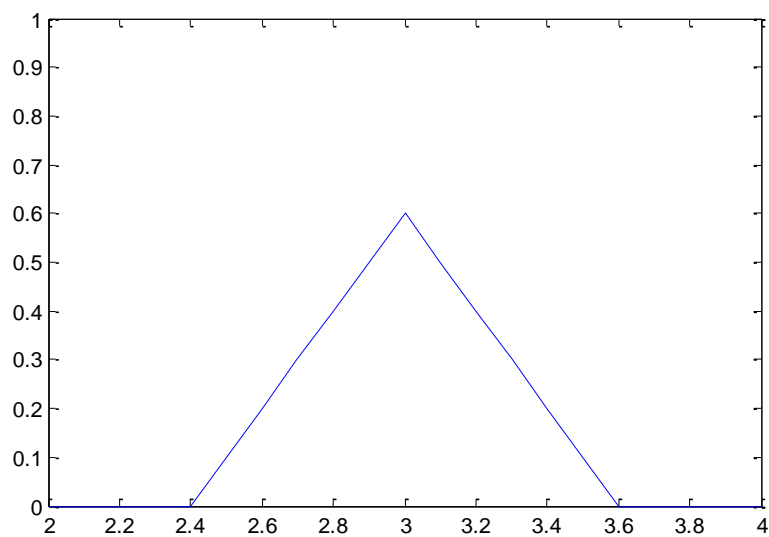
شکل 5-1: استنتاج bounded sum - dot

ث) bounded sum به جای S_n و min به جای T_n : شکل 6-1 نتیجه استنتاج را نشان می‌دهد:



شکل 6-1: استنتاج bounded sum - min

ج) bounded sum به جای S_n و bounded difference به جای T_n : شکل 7-1 نتیجه استنتاج را نشان می‌دهد:



شکل 7-1 استنتاج bounded sum – bounded difference

2. روش های غیر فازی سازی را که می شناسید توضیح دهید.

پارامترهای کنترلی که به فرایند اعمال می کنیم بصورت اعداد و مقادیر قطعی و غیر فازی هستند ولی مقادیر بدست آمده از موتور استنتاج، مقادیری فازی بوده و نیاز داریم آن مقادیر یا مجموعه های فازی را به مقادیر غیر فازی مناسب تبدیل نماییم.

ورودی بلوک غیر فازی کننده، خروجی موتور استنتاج و اجتماع نتایج بدست آمده از قاعده های درگیر در رقابت است. هر قاعده، خروجی C'_i را داشته که ارتفاع آن α_i - قدرت آن قاعده را در رقابت مشخص می نماید و مقدار قله C'_i برابر مقداری از Z است که در آن C'_i بیشترین مقدار را داشته است و آنرا با $Z^{(i)}$ نمایش می دهیم. به عنوان مثال اگر C'_i یک تابع عضویت مثلثی باشد، رأس مثلث بیشترین میزان عضویت را داراست و نقطه ای از محور Z که در آن نقطه تابع عضویت بیشترین مقدار را دارد مقدار قله یا $Z^{(i)}$ نام دارد. اگر تابع عضویت دوزنقه ای باشد، میانگین نقاطی که دارای میزان عضویت بیشینه هستند به عنوان مقدار قله فرض می شود. خروجی Z^* بعنوان خروجی بلوک غیر فازی کننده می تواند بوسیله یکی از روشهای زیر بدست آید:

روش مرکز ثقل

این روش تقریباً بهترین روش غیر فازی کردن است در حالیکه فضای تعریف گسسته باشد و $Z = \{z_1, z_2, \dots, z_n\}$ مقدار فازی Z^* بصورت زیر بدست می آید:

$$Z^* = \sum z_i \mu_c(z_i) / \sum \mu_c(z_i) = \sum z_i \text{MAX} \mu_{c'k}(z_i) / \sum \text{MAX} \mu_{c'k}(z_i)$$

و اگر فضای Z پیوسته باشد:

$$Z^* = \int z \mu_c(z) dz / \int \mu_c(z) dz = \int z \text{MAX} \mu_{c'k}(z) dz / \int \text{MAX} \mu_{c'k}(z) dz$$

در این روش، غیر فازی کردن از لحاظ محاسباتی پیچیده است و سرعت کم از معایب کنترل کننده ای است که از این نوع غیر فازی کننده استفاده می کند.

روش مرکز مجموع ها

در روش مرکز ثقل مجبور بودیم تابع عضویت خروجی C را محاسبه نماییم تا در صورتیکه دو C'_i همپوشانی داشتند ناحیه همپوشانی را فقط یکبار در محاسبات وارد نماییم اما در این روش هر C'_i به صورت جداگانه ای محاسبه می شود و این امکان را به ما می دهد که بدون محاسبه C از سرعت بیشتری برخوردار شویم. این الگوریتم یکی از سریعترین روشهای غیرفازی کردن است و به همین دلیل بیشتر کنترل کننده های فازی از این الگوریتم استفاده می نمایند. نحوه انجام این روش برای فضای تعریف گسسته و پیوسته به ترتیب زیر است:

$$Z^* = \sum z_i \sum \mu_{c'k}(z_i) / \sum \sum \mu_{c'k}(z_i)$$

$$Z^* = \int z \sum \mu_{c'k}(z) dz / \int \sum \mu_{c'k}(z) dz$$

روش ارتفاع

در این روش نیز به محاسبه C نیازی نداریم و می توانیم از خروجی های مجزا بریده شده یا مقیاس شده استفاده نماییم. این روش مقدار قله هر C'_k را با وزن α_k در نظر می گیرد و برای سیستمی که دارای n قاعده است خروجی قطعی به صورت زیر بدست می آید:

$$Z^* = \sum \alpha_k \cdot z^{(k)} / \sum \alpha_k$$

این روش بسیار ساده و سریع است.

روش مرکز بزرگترین سطح

از این روش زمانی استفاده می شود که خروجی C بصورت یک تابع عضویت غیر محدب باشد. به عبارت دیگر، تابع عضویت خروجی C همگرا نباشد و حداقل شامل دو زیر مجموعه فازی همگرا باشد. در این حالت زیر مجموعه های فازی همگرا را مشخص می نماییم و

سطح زیر نمودار هر کدام از آنها را نیز به دست می آوریم. مقدار قطعی Z^* را از روش مرکز ثقل برای زیر مجموعه فازی همگرایی که بیشترین سطح را داشته باشد به دست می آوریم. بدیهی است در صورتیکه تابع عضویت خروجی Z محدب باشد این روش به روش مرکز ثقل تبدیل می شود.

روش متوسط ماکزیمم

این روش در سه مرحله انجام می شود و نهایتاً متوسط مقادیری از Z که دارای بیشترین مقدار عضویت در C هستند به عنوان Z^* انتخاب می شود.

مرحله 1: درجه عضویت ماکزیمم در خروجی c محاسبه می شود:

$$\text{hgt}(Z) = \text{MAX } \mu_c(z)$$

مرحله 2: مجموعه ای از اعضا Z که دارای درجه عضویت $\text{hgt}(z)$ هستند، تشکیل می گردد:

$$M = \{z | z \in Z, \mu_c(z) = \text{hgt}(Z)\}$$

مرحله 3:

$$Z^* = (\text{Inf } M + \text{Sup } M) / 2$$

ممکن است تابع عضویت Z^* برابر $\text{hgt}(Z)$ نباشد که در این صورت باید بسته ترین فاصله با ماکزیمم تابع عضویت را برای M بیابیم. علاوه بر روش متوسط ماکزیمم، روشهای اولین ماکزیمم و آخرین ماکزیمم نیز وجود دارند که کوچکترین و بزرگترین عضو مجموعه M را به عنوان مقدار قطعی Z^* در نظر می گیرند.

غیرفازی کننده در مدل ساگنو

در مدل ساگنو خروجی کنترل کننده به صورت تابعی از ورودی ها در نظر گرفته می شود و بعنوان مثال:

Rule i : if x is A_i and y is B_i then z is $F(x, y)$

در این حالت برای بدست آوردن خروجی قطعی Z^* از رابطه زیر استفاده می کنیم:

$$Z^* = \sum \alpha_i . F_i(x, y) / \sum \alpha_i$$

که این روش شباهت زیادی به روش ارتفاع دارد.

3. ساختار کنترلر فازی و روشهای مختلف ساخت آنرا توضیح دهید؛ روشهای SOC, Adaptive را با ذکر جزئیات بیان کنید.

کنترل فازی و کنترل کلاسیک دارای شباهت ها و تفاوت هایی می باشند. این دو در موارد زیر مشابه می باشند:

- هر دو سعی در حل نوع یکسانی از مسائل را دارند که همان مسائل کنترل می باشد، بنابراین باید به جنبه های یکسانی که در هر مسأله کنترل مشترک می باشد توجه داشته باشند؛ بعنوان مثال: پایداری و عملکرد.
- ابزارهای ریاضی مورد استفاده جهت تحلیل سیستم های کنترل طراحی شده، مشابه می باشند چرا که به مطالعه جنبه های یکسانی برای یک سیستم می پردازند.

به هر حال یک تفاوت اساسی بین کنترل فازی و کنترل کلاسیک وجود خواهد داشت:

- کنترل کلاسیک جهت طراحی کنترل کننده با یک مدل ریاضی از فرایند شروع می نماید و کنترل کننده ها برای مدل طراحی می شوند، درحالیکه در کنترل فازی با استفاده از تجربیات شخص خبره (بصورت قواعد اگر – آنگاه فازی) شروع می گردد و کنترل کننده ها بوسیله ترکیب این قواعد طراحی می گردند. بدین معنی که اطلاعات لازم جهت ساخت این دو کنترل کننده فازی متفاوت است. کنترل کننده های فازی می توانند با استفاده از هر دو مدل ریاضی و هیوریستیک ساخته شوند.

برای بسیاری از مسائل کنترل عملی (مثلا کنترل فرایند صنعتی) مشاهده یک مدل ریاضی ساده و در عین حال دقیق مشکل می باشد، اما می تواند آزمایشاتی توسط یک شخص ماهر و باتجربه فراهم شود که یک راهکار عملی و تجربی مفید برای کنترل نمودن ارائه نماید. کنترل فازی برای این دو نوع مسائل بیش از هر راه دیگری مفید می باشد.

روشهای طراحی برای کنترل کننده های فازی را به دو دسته تقسیم بندی می کنیم، رهیافت سعی و خطا و روش تئوریک. در رهیافت سعی و خطا، با استفاده از دانشی که بر پایه آزمایشات و تجربیات بدست آمده و به صورت شفاهی قابل بیان است (مثلا مجموعه اعمال دستی و اپراتوری)، همچنین با سوالهایی که از خبرگان مربوطه صورت می گیرد و پرسشنامه های دقیقی تنظیم می شود، مجموعه ای از قواعد اگر – آنگاه فازی جمع آوری می گردد و آنگاه در ادامه، کنترل کننده های فازی بر اساس این قواعد ساخته شده و در نهایت در سیستم مورد آزمایش قرار می گیرند. حال اگر بکارگیری کنترل کننده فازی طراحی شده در عمل مورد رضایت نباشد، قواعد مجددا به صورت مناسبی تغییر و تنظیم می گردند و یا دوباره از ابتدا ایجاد می شوند و این کار آنقدر ادامه می یابد تا پس از چند چرخه سعی و خطا، عملکرد نهایی رضایتبخش گردد.

در روش تئوریک، ساختار و پارامترهای کنترل کننده های فازی چنان طراحی می شوند که معیار عملکرد مشخصی (برای مثال پایداری) تضمین گردد. البته بهتر است که در طراحی کنترل کننده های فازی برای سیستم های عملی، هر دو روش را ترکیب نماییم تا امکان دستیابی به بهترین کنترل کننده فازی فراهم آید.

رهیافت سعی و خطا جهت طراحی کنترل کننده های فازی می تواند در سه گام زیر خلاصه شود:

گام اول: تحلیل سیستم حقیقی و انتخاب متغیرهای حالت و کنترل.

متغیرهای حالت باید خصوصیات و ویژگی های اصلی و کلیدی سیستم را توصیف نمایند و متغیرهای کنترلی باید قادر به تأثیر گذاری بر حالت های سیستم باشند. متغیرهای حالت ورودی کنترل کننده فازی و متغیرهای کنترل، خروجی کنترل کننده فازی خواهند بود.

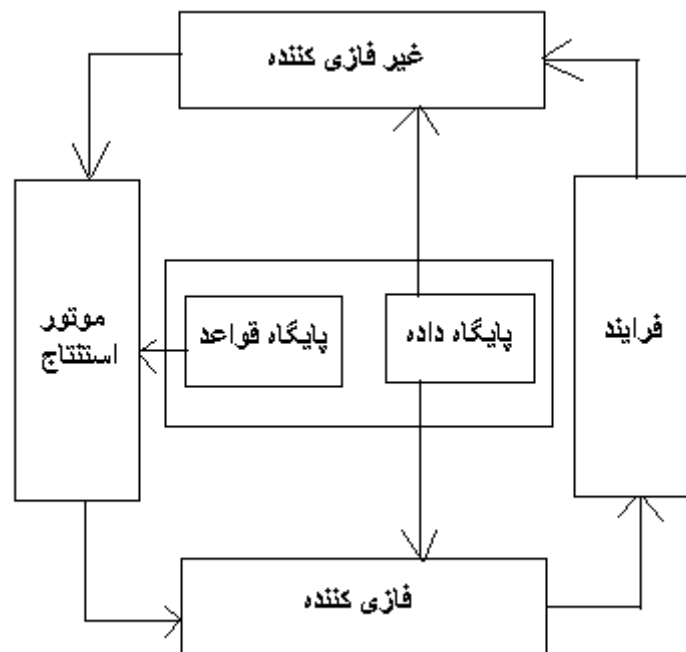
گام دوم: استخراج قواعد اگر – آنگاه فازی که متغیرهای حالت را به متغیرهای کنترل مرتبط نماید.

فرموله نمودن این قواعد می تواند با استفاده از دو روش تجربی انجام گیرد. روش عمومی تر از تجربیات خبرگان که بصورت شفاهی بیان می شود استفاده می نماید. روش دیگر شامل تنظیم پرسشنامه ای دقیق از تجربیات کارشناسان می باشد. با این راهها می توانیم نخستین نمونه از قواعد کنترل فازی را بدست آوریم.

گام سوم: ترکیب قواعد اگر – آنگاه فازی استخراج شده در یک سیستم فازی و آزمایش سیستم حلقه بسته با استفاده از این سیستم فازی به عنوان کنترل کننده.

این گام بدین معنی است که سیستم حلقه بسته با کنترل کننده فازی راه اندازی گردد و در صورتیکه عملکرد سیستم مورد رضایت نباشد، سعی گردد تا کنترل کننده فازی به خوبی تنظیم شود و یا مجددا کنترل کننده فازی بوسیله سعی و خطا طراحی گردد و عملیات تکرار شود تا عملکرد رضایت بخش شود.

ساختار کلی یک کنترل کننده فازی بدین صورت است:

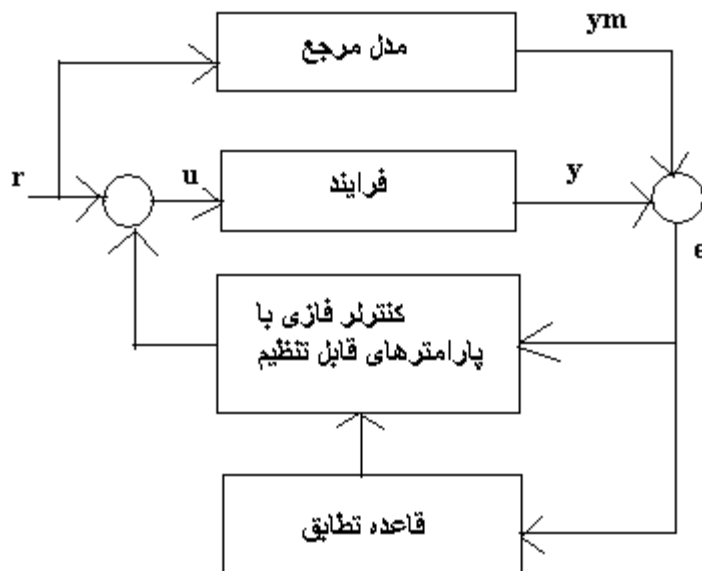


شکل 3- 1، ساختار کلی یک کنترلر فازی

همانگونه که در شکل فوق مشاهده می شود، بلوکهای فازی کننده و غیرفازی کننده برای ارتباط با فرایند تحت کنترل به کار گرفته شده اند زیرا در فرایند، مقادیر یا پارامترهای تحت کنترل، به صورت دقیق و قطعی توسط سنسورها دریافت می شوند و به یک فازی کننده نیاز داریم تا آن پارامترها را به قالبی ببرد که در موتور استنتاج قابل استفاده باشد. و همچنین هنگام اعمال نتیجه موتور استنتاج از آنجا که بوسیله متغیرهای قطعی می توانیم بر فرایند اعمال نفوذ نماییم لازم است تا توسط یک غیر فازی کننده، نتیجه فازی موتور استنتاج به یک مقدار قطعی تبدیل شود.

کنترل کننده های فازی در وضعیت هایی کار می کنند که در آن یک عدم قطعیت بزرگ با تغییرات نامعلوم در پارامترها و ساختارهای سیستم وجود دارد. عموماً هدف اصلی کنترل تطبیقی عبارت از ثابت نگه داشتن کارایی یک سیستم در حضور این عدم قطعیت هاست. بنابراین کنترل فازی پیشرفته باید تطبیقی باشد.

ساختار اصلی یک سیستم کنترل فازی تطبیقی در شکل زیر نشان داده شده است.



شکل 3-2، کنترلر فازی تطبیقی

مدل مرجع به منظور مشخص نمودن پاسخ ایده آل که سیستم کنترل فازی بایستی پیروی نماید، استفاده می گردد. فرض می شود که سیستم تحت کنترل دارای اجزایی ناشناخته باشد. کنترل کننده های فازی از سیستم های فازی ساخته می شوند که در آن پارامترهای θ قابل تنظیم می باشند. قاعده تطابق، پارامترهای θ را بصورت برخط چنان تنظیم می کند که خروجی سیستم تحت کنترل یعنی $y(t)$ خروجی مدل مرجع یعنی $y_m(t)$ را دنبال نماید. مشاهده می شود که تفاوت اصلی بین سیستم های کنترل فازی تطبیقی و سیستم های کنترل فازی غیر تطبیقی عبارتند از: (i) کنترل کننده های فازی در سیستم کنترل فازی تطبیقی در طی عملیات زمان حقیقی تغییر می کنند در حالی که کنترل کننده فازی در سیستم کنترل فازی غیر تطبیقی قبل از عملیات زمان حقیقی مشخص و ثابت خواهند بود، و (ii) یک جزء اضافی به نام قاعده تطابق به سیستم کنترل فازی تطبیقی اضافه می شود تا پارامترهای کنترل کننده فازی را تنظیم سازد. مزیت های اصلی کنترل فازی تطبیقی نسبت به کنترل کننده فازی غیر تطبیقی عبارتند از: (i) عملکرد و کارایی بهتر معمولاً قابل دستیابی می باشد، چرا که کنترل کننده فازی تطبیقی می تواند خود را با توجه به تغییرات محیطی تنظیم نماید و (ii) دانش کمتری از سیستم تحت کنترل لازم می باشد چون قاعده تطابق می تواند در جهت یادگیری دینامیک سیستم در طی عملیات زمان حقیقی کمک نماید.

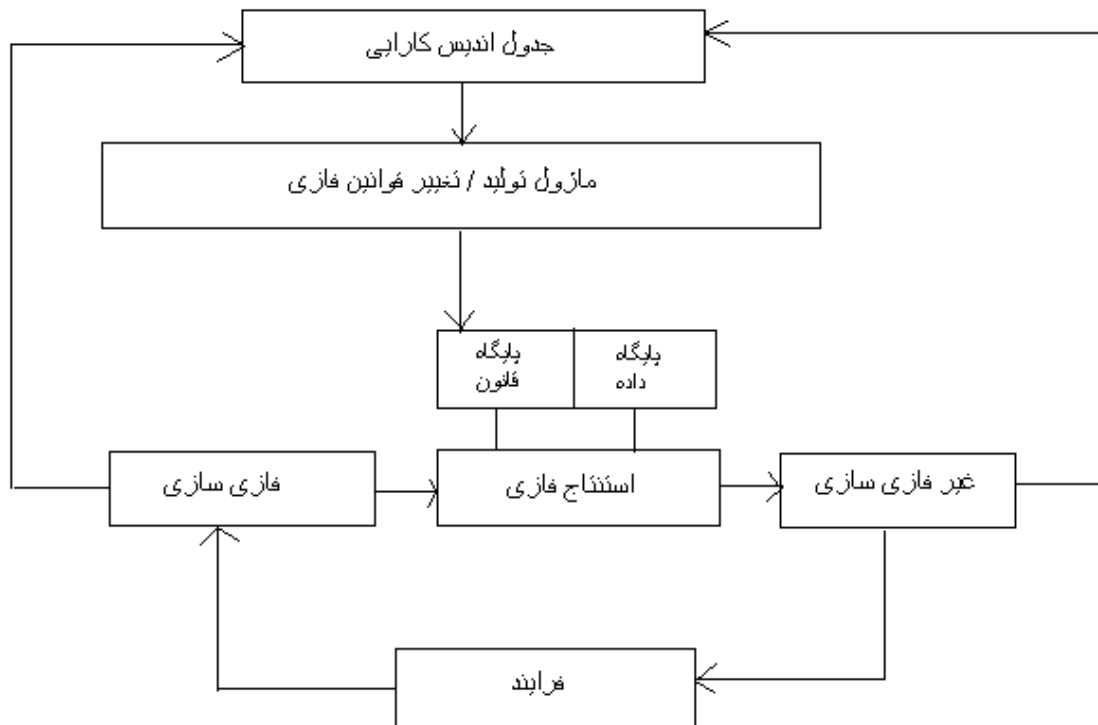
معایب اصلی کنترل فازی تطبیقی نسبت به کنترل فازی غیر تطبیقی عبارتند از (i) تحلیل نتایج سیستم کنترل مشکل است زیرا سیستم کنترل نه تنها غیر خطی بلکه متغیر با زمان نیز می باشد و (ii) پیاده سازی آن گران و پرهزینه می باشد.

یک FLC خود تنظیم یا SOC، یک نوع کنترلر دو – سطحی، سلسله مراتبی و مبتنی بر قانون می باشد که در آن، پایگاه قانون کنترلر توسط یک مازول یادگیری و با الگوریتم خود – تنظیم در هنگام اجرای وظایف کنترلی ایجاد شده یا تغییر می یابد. همانگونه که در شکل زیر نشان داده شده است، یک FLC خود تنظیم از قسمت های زیر تشکیل شده است:

1. یک کنترلر فازی معمولی در سطح پایین.

2. یک مازول یادگیری در سطح بالا.

ماژول یادگیری که شامل یک جدول اندیس کارایی و یک الگوریتم تولید و تغییر قانون می باشد، مسئول ایجاد قوانین جدید یا تغییر قوانین موجود است. در یک FLC خود - تنظیم حالت کنترلی یک فرایند توسط ماژول یادگیری مشاهده می شود. وقتی خروجی نامطلوبی از فرایند تشخیص داده شود، قوانین کنترل فازی بر اساس تصحیحاتی که از جدول اندیس کارایی بدست می آید، ایجاد شده یا تغییر می یابند. جدول اندیس کارایی باید قبل از هر جزئی در یک FLC خود - تنظیم، طراحی گردد.



شکل 3-3، کنترلر فازی SOC

جدول اندیس کارایی، حالت فرایند را با انحراف از رفتار مطلوب مرتبط می سازد و تصحیحاتی را که لازم است تا FLC، سیستم را به حالت مطلوب ببرد، ایجاد می کند. بسته به ساختار یک FLC، یک جدول اندیس کارایی می تواند یا بطور زبانی تعریف گردد یا با استفاده از اندیس های کارایی مثل خطا، میانگین مربعات خطا، بزرگترین خطای مطلق و خطای میانگین متغیرهای سیستم، بصورت کمی بیان شود.

جدول اندیس کارایی به موارد زیر بستگی دارد:

1. مقادیر زبانی و سطوح تدریج بکار رفته.

2. ساختار قواعد فازی..

معمولا جدول اندیس کارایی از روی یک مجموعه کلی از قوانین زبانی استخراج می گردد که خط سیر کنترلی مطلوب در فضای حالت سیستم را بیان می کنند.

4. روشهای کلاسترینگ فازی را که می شناسید توضیح دهید.

در بازشناسی الگو، گروهی از داده ها را یک خوشه می نامند. در عمل، داده ها معمولاً بخوبی توزیع نشده اند، بنابراین "قواعد" و "ساختارها" را ممکن است نتوان بدقت تعریف نمود. بدین معنی که بازشناسی الگو فی النفسه یک علم نادقیق است. در کار با این حوزه مبهم، دخالت دادن "فازی بودن" در فرموله کردن مسائل مفید خواهد بود. بعنوان مثال مرز بین خوشه ها به جای قطعی و مشخص بودن می تواند فازی باشد، بدین معنی که یک نقطه داده می تواند به دو یا چند خوشه و با درجه های مختلف تعلق داشته باشد. در این صورت، فرموله سازی به دنیای واقعی نزدیکتر بوده و بنابراین انتظار کارایی بیشتری از آن می توان داشت. این اولین دلیل برای استفاده از مدل های فازی جهت بازشناسی الگو می باشد: مسأله بدلیل طبیعتش نیازمند مدل سازی فازی است، در حقیقت مدل سازی فازی، مدل سازی انعطاف پذیرتری می باشد؛ با توسعه تعلق صفر – یک به تعلق در محدوده $[0, 1]$ انعطاف پذیری بیشتری حاصل می شود.

دلیل دوم برای استفاده از مدل های فازی این است که مسائل فرموله شده از نظر محاسباتی راحت تر حل می شوند. این بواسطه این واقعیت است که یک مدل غیر فازی اغلب بوسیله جستجوی کامل در یک فضای بسیار بزرگ نتیجه می دهد چون تعداد زیادی از متغیرهای کلیدی فقط می توانند دو مقدار صفر و یک را اختیار کنند درحالیکه در یک مدل فازی، تمامی متغیرها پیوسته بوده و در نتیجه برای پیدا کردن مسیر صحیح جستجو می توان از مشتق استفاده کرد. مسأله اصلی در بازشناسی الگو، پیدا کردن خوشه ها از مجموعه نقاط داده می باشد.

الگوریتم c – means فازی

گام اول: برای مجموعه داده $X = \{x_1, x_2, \dots, x_n\}$ و $x_i \in \mathbb{R}^p$ ، مقدار $c \in \{2, 3, \dots, n-1\}$ و $m \in (1, \infty)$ را مشخص کرده و $U^{(0)} \in M_{fc}$ را مقداردهی اولیه نمایید.

گام دوم: در مراحل تکرار 1 که $l = 0, 1, 2, \dots$ ، بردارهای $c - \text{means}$ را محاسبه نمایید.

$$v^{(l)}_i = \sum (u^{(l)}_{ik})^m x_k / \sum (u^{(l)}_{ik})^m \quad 1 \leq i \leq c$$

گام سوم: مقدار $U^{(l)} = [u^{(l)}_{ik}]$ را به $U^{(l+1)} = [u^{(l+1)}_{ik}]$ با استفاده از رابطه زیر بهنگام کنید.

$$U^{(l+1)}_{ik} = 1 / (\sum [\|x_k - v^{(l)}_i\| / \|x_k - v^{(l)}_j\|]^{(2/(m-1))}) \quad 1 \leq i \leq c, 1 \leq k \leq n$$

گام چهارم: اگر $\|U^{(l+1)} - U^{(l)}\| < \varepsilon$ کار را متوقف کنید، در غیر اینصورت $l = l+2$ قرار داده و به گام دوم بروید.

5. عدد فازی را تعریف کرده برای دو عدد فازی A, B مقادیر $A.B = C$, $C / B = D$ را با دو روش $\alpha - \text{cut}$ و اصل تعمیم محاسبه نمایید. آیا $D = A$ است؟

مفهوم عدد فازی از این واقعیت نشأت گرفته که بسیاری از پدیده های کمی قابل ارائه با یک عدد مطلق و بدون ابهام نیستند. مثلا اغلب انسانها دارای ساعتی هستند که تا حدودی غیر دقیق بوده و اگر وقت را از آنها بپرسید، بعنوان مثال می گویند: ساعت الان "حدود دو بعد از ظهر است". یا ممکن است یک فرد خود را محدود به یک برنامه در زمان کاملا دقیق ننموده و یک دعوت را با زمان "حدودا شش و سی دقیقه" بیان کند. در یک مغازه میوه فروشی، اگر یک دسته موز "تقریبا چهارپوند" باشد، ما راضی هستیم. بنابراین عددی فازی است که بتوان آنرا در قالب یک صفت زبانی و یا لغت عددی مانند تقریبا، حدودا، نزدیک به و غیره بیان کرد.

مشاهده می شود که مفهومی از بیان "تقریبا شش" به ذهن متبادر می شود، یک مفهوم فازی بوده زیرا شامل مقادیر عددی بین دو طرف مقدار مرکزی شش می باشد. اگر مقدار مرکزی کاملا با این مفهوم منطبق بود، اعداد حول این مرکز نیز با درجات پایین تر با آن انطباق دارند. بدیهی است که درجه انطباق هر عدد با مفهوم مورد نظر، با نزدیکی به مقدار مرکزی قابل بیان است. بعبارت دیگر، هر مفهوم با یک مجموعه فازی تعریف شده در مجموعه اعداد حقیقی قابل اتخاذ است. در این صورت، تابع عضویت برای مقدار مرکزی مساوی یک و برای سایر اعداد، بسته به نزدیکی آنها به مقدار مرکزی بین صفر و یک خواهد بود. در این صورت، تابع عضویت باید از یک به صفر برای دو طرف مقدار مرکزی کاهش یابد. چنین مجموعه های فازی را اعداد فازی گویند.

اعداد فازی حائز نقش بسیار مهمی در عمل مانند تصمیم گیری، استدلال تقریبی، کنترل فازی و آمار و احتمالات غیر صریح دارند. مثلا تصمیم گیری در مورد بازار سهام را می توان بصورت زیر تصور نمود:

"اگر قیمت یک سهام خاص به تقریبا 50 دلار برسد، مدیر مالی بایستی تقریبا نصف سهام خود را بفروش برساند".

در حالیکه هر عدد فازی مانند A از طریق تابع عضویت آن بصورت زیر بیان می گردد:

$A: R \rightarrow [0, 1]$ ، با این حال همه توابع عضویت به شکل بالا مبین یک عدد فازی نیستند.

برای اینکه یک تابع عضویت متعلق به یک عدد فازی باشد بایستی آن تابع عضویت مفهوم مجموعه ای از اعداد که در اطراف یک عدد حقیقی و یا اطراف یک بازه از اعداد حقیقی قرار گرفته را بیان و تفسیر نماید. توابع عضویتی که دارای مشخصه بالا باشد را می توان بصورت زیر فرموله نمود:

$$A(x) =$$

$$\begin{aligned} f(x) & \quad x \in [a, b] \\ 1 & \quad x \in [b, c] \\ g(x) & \quad x \in [c, d] \\ 0 & \quad x < a, x > d \end{aligned}$$

در این فرمول، $a \leq b \leq c \leq d$ ، f یک تابع پیوسته که از نقطه b تا 1 افزایشی بوده و y یک تابع پیوسته کاهشی از 1 تا نقطه c می باشد.

هر عدد فازی باید دارای ویژگی های زیر باشد:

1. نرمال باشد: هسته (core) آن مجموعه تهی نباشد.
2. مقاطع α آن بازه های بسته اعداد حقیقی باشند.
3. دارای پشتیبان در یک بازه (a, d) از اعداد حقیقی اند.
4. محدب اند.

این ویژگی ها برای تعریف معنی دار عملیات حسابی در اعداد فازی بسیار مهم و اساسی اند. بدلیل اینکه هر مجموعه فازی قابل نمایش بصورت مقاطع α آن بوده و این مقاطع بازه های بسته اعداد حقیقی اند، عملیات حساب روی اعداد فازی را می توان بصورت این عملیات روی بازه های اعداد حقیقی تعریف نمود. عملیات روی بازه ها بخش جالبی از ریاضیات کلاسیک اند.

عمل ضرب و تقسیم روی دو بازه بدین صورت انجام می گیرد:

$$[a, b].[c, d] = [\min(ac, ad, bc, bd), \max(ac, ad, bc, bd)]$$

$$[a, b] / [c, d] = [\min(a/c, a/d, b/c, b/d), \max(a/c, a/d, b/c, b/d)]$$

جهت استفاده مفید از اعداد فازی بایستی چهار عمل اصلی را به این اعداد بسط داد. بدلیل قابلیت انتقال اعداد فازی به مقاطع α ، عملیات روی بازه ها، قابل انتقال به عملیات روی اعداد فازی هستند. دو عدد فازی A, B را در نظر گرفته و فرض کنید علامت ستاره (*) یکی از چهار عمل اصلی روی بازه ها باشد. بنابراین برای هر $\alpha \in [0, 1]$ ، مقطع α برای $A*B$ بصورت مقاطع α برای A, B و با فرمول زیر بیان می گردد:

$$\alpha_{(A*B)} = \alpha_A * \alpha_B$$

هنگامیکه مقطع α بصورت $\alpha_{(A*B)}$ تعیین شد، نتیجه عدد فازی $A*B$ بصورت زیر تعیین می گردد:

$$A*B = U \alpha_{(A*B)}. \alpha$$

اصل توسعه یک معادله اساسی است که اجازه می دهد دامنه یک تابع را از نقاط در U به مجموعه های فازی در U توسعه داد. مشخص تر این که فرض کنید $f: U \rightarrow V$ تابعی از مجموعه قطعی U به مجموعه قطعی V باشد. همچنین فرض کنید که یک مجموعه فازی A در U داده شده و ما می خواهیم مجموعه فازی B را در V به نحوی معین کنیم که $B = f(A)$. اگر f یک نگاشت یک به یک باشد آنگاه داریم:

$$\mu_B(y) = \mu_A[f^{-1}(y)], y \in V$$

اگر f یک به یک نباشد، آنگاه هنگامی که دو یا چند نقطه متمایز در U با مقادیر تعلق متفاوت در A به یک نقطه یکسان در V نگاشته می شوند، ابهامی بوجود خواهد آمد. برای رفع این ابهام، مقدار بزرگتر را از بین دو مقدار تعلق به $\mu_B(y)$ نسبت می دهیم. در حالت کلی، تابع تعلق برای B بدین صورت تعریف می شود:

$$\mu_B(y) = \max_{x \in f^{-1}(y)} \mu_A(x), y \in V$$

این معادله اصل توسعه نامیده می شود که می توان از آن برای انجام عملیات حسابی روی اعداد فازی استفاده کرد، بدین صورت که :

$$\mu_{A.B}(z) = \sup \min[\mu_A(x), \mu_B(y)] , xy = z$$

$$\mu_{A/B}(z) = \sup \min[\mu_A(x), \mu_B(y)] , x/y = z$$

اثبات می شود که استفاده از روش اصل توسعه و روش مقاطع α برای اعمال حسابی روی اعداد فازی نتیجه یکسانی بدست می دهد.

برای محاسبه حاصلضرب 2 عدد فازی A, B می توانیم از مقاطع آلفای آنها استفاده کنیم برای این منظور فرض می کنیم مقطع $\alpha - A, B$ به ترتیب $[c, d]$ و $[a, b]$ باشد، آنگاه داریم:

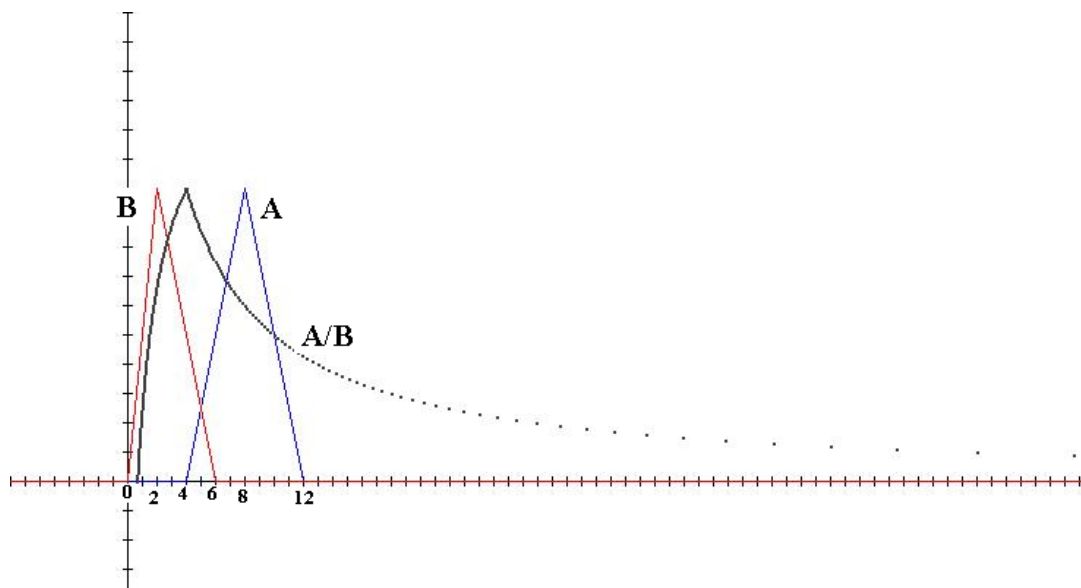
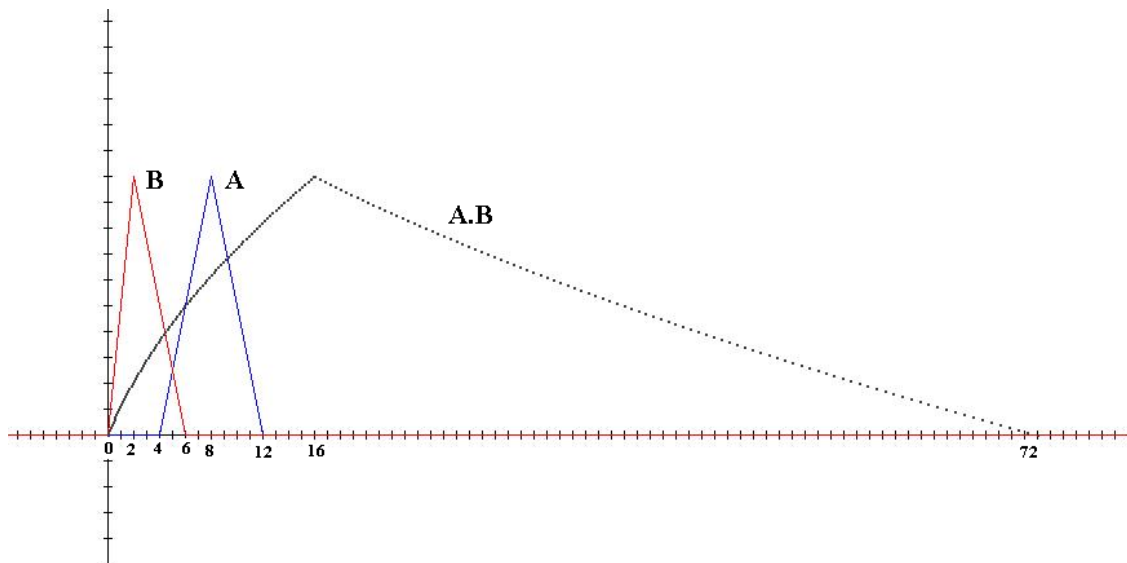
$$[a, b] * [c, d] = [\min(ac, ad, bc, bd), \max(ac, ad, bc, bd)]$$

با یک برنامه ساده می توان به ازای $0 \leq \alpha \leq 1$ ، مقاطع آلفا را بدست آورده و به ازای هر کدام از آنها دو نقطه یعنی نقاط انتهایی بازه حاصلضرب که از روش بالا بدست می آید را حساب کرد. با وصل کردن این نقاط به یکدیگر، نمودار حاصلضرب فازی بدست می آید.

عمل تقسیم هم مشابه عمل ضرب است با این تفاوت:

$$[a, b] / [c, d] = [\min(a/c, a/d, b/c, b/d), \max(a/c, a/d, b/c, b/d)]$$

نتیجه این دو عمل با برنامه ای که نوشته شد، بدین صورت می باشد:

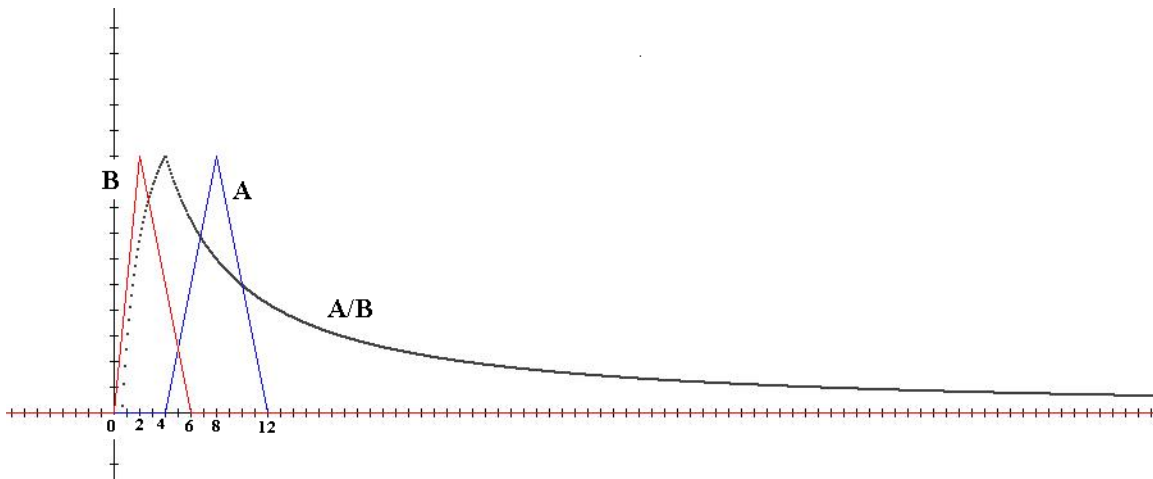
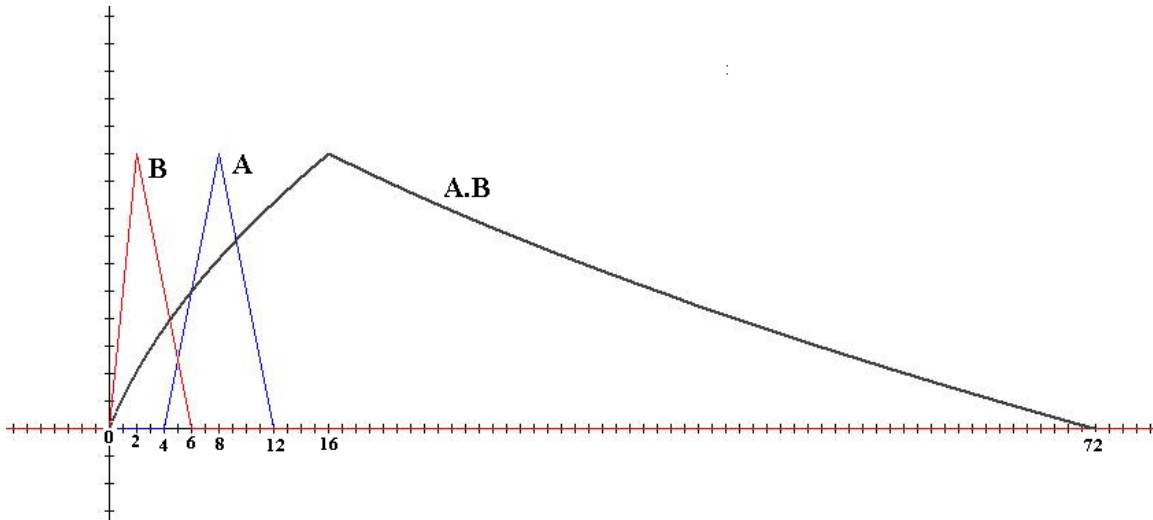


همچنین اعداد فازی را می توان با روش اصل توسعه در هم ضرب یا بر هم تقسیم کرد. در این روش اعداد بصورت زیر در هم ضرب یا بر هم تقسیم می شوند:

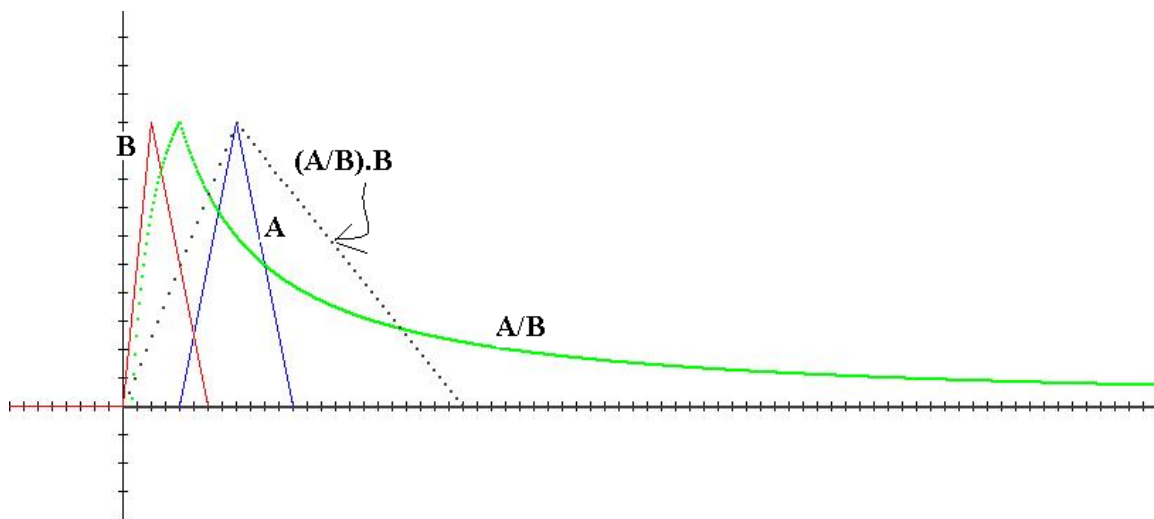
$$\mu_{A.B}(z) = \sup \min[\mu_A(z), \mu_B(z)] \text{ for all } x, y \text{ that } x.y = z$$

$$\mu_{A/B}(z) = \sup \min[\mu_A(z), \mu_B(z)] \text{ for all } x, y \text{ that } x/y = z$$

باز هم با یک برنامه ساده می توان این ضرب و تقسیم را انجام داد در این قسمت به جای اینکه روی α حرکت کنیم، روی محور x یا y حرکت می کنیم و از روی آنها مقادیر تابع عضویت حاصلضرب یا حاصل تقسیم را بدست می آوریم.



حال می خواهیم مقایسه کنیم که آیا $(A/B).B = A$ می باشد یا نه؟ با استفاده از همان برنامه قبلی این تست را انجام می دهیم:



ملاحظه می کنیم که این تساوی برقرار نمی باشد.

6. روش مدلینگ TSM را توضیح دهید.

در برخی سیستم های کنترل فازی، فرایند کنترل بوسیله مدل های دینامیکی سیستم خطی یا غیرخطی عادی ارائه می گردد. در بسیاری از مسائل عملی، اشخاص خبره می توانند توصیف های زبانی (بصورت قواعد اگر – آنگاه فازی) در رابطه با فرایند را فراهم نمایند که می تواند با مدلی از فرایند ترکیب گردد که این مدل، مدل سیستم فازی نامیده می شود. بنابراین موضوع جالب، مطالعه سیستم کنترل فازی در حالتی است که فرایند بوسیله سیستم های فازی مدل شده و کنترل کننده پس خورد یک کنترل کننده فازی می باشد.

سیستم فازی TSM بعنوان نوع دیگری از سیستم های فازی پیشنهاد گردیده است و از قواعدی به شکل زیر بنا شده است:

$$\text{اگر } x_1, \dots, x_n \text{ است و } C^l_n \text{ است، آنگاه } y^l = c^l_0 + c^l_1 x_1 + \dots + c^l_n x_n$$

که C^l_i مجموعه های فازی، c^l_i ضرایب ثابت و $l = 1, 2, \dots, M$ می باشند. در این رابطه، قسمت های اگر قواعد همانند قواعد اگر – آنگاه سیستم های فازی معمولی می باشند، اما قسمت های آنگاه، ترکیب خطی از متغیرهای ورودی هستند. به ازای یک ورودی $x = (x_1, \dots, x_n)^T \in U$ خروجی $f(x) \in V$ سیستم فازی TSM بصورت متوسط وزندار شده y^l ها محاسبه می گردد یعنی:

$$f(x) = \sum y^l w^l / \sum w^l$$

که وزنه های w^l از رابطه زیر محاسبه می شوند:

$$w^l = \pi \mu C^l_i(x_i)$$

مشاهده می شود که سیستم فازی TSM هنوز بعنوان یک نگاشت از U به V مطرح می گردد. معنی فیزیکی آن این است که وقتی x به محدوده فازی مقید می گردد که بوسیله بخش اگر قاعده توصیف می گردد، خروجی یک تابع خطی از متغیرهای ورودی می باشد. بنابراین سیستم فازی می تواند تا اندازه ای همانند یک تابع قطعه قطعه خطی تصور گردد که تغییرات آن از یک قطعه به قطعه دیگر ملایم می باشد. اگر $c^l_i = 0$ برای $i = 1, 2, \dots, n$ و c^l_0 معادل با مرکز y^l از مجموعه فازی B^l در قاعده اگر – آنگاه فازی باشد، آنگاه سیستم فازی TSM با سیستم فازی که

بوسیله موتور استنتاج ضرب، فازی ساز منفرد و غیر فازی ساز میانگین مراکز تعریف می گردد، یکسان خواهد بود.

اگر خروجی سیستم فازی TSM بعنوان یکی از ورودیهای آن بکار رود، آنگاه سیستمی بدست می آوریم که سیستم فازی دینامیک TSM نامیده می شود. بطور مشخص، سیستم فازی TSM دینامیک از قواعدی به شکل زیر بنا خواهد شد:

$$\text{اگر } x(k), A^p_1 \text{ و } \dots \text{ و } x(k-n+1), A^p_n \text{ و } u(k), B^p \text{ است آنگاه}$$

$$x^p(k+1) = a^p_1 x(k) + \dots + a^p_n x(k-n+1) + b^p u(k)$$

که A^p_i و B^p مجموعه های فازی، a^p_i و b^p برای $p = 1, 2, \dots, N$ ثابتها، $u(k)$ ورودی سیستم و $X(k) = (x(k), x(k-1), \dots, x(k-n+1))^T$ بردار حالت سیستم می باشد. خروجی سیستم فازی دینامیکی TSM بوسیله رابطه

$$x(k+1) = \sum x^p(k+1) v^p / \sum v^p$$

محاسبه می شود و v^p بصورت زیر است:

$$v^p = \pi \mu_{A^p_i} [x(k-i+1)] \mu_{B^p} [u(k)]$$

از سیستم فازی TSM دینامیک در مدل سازی فرایند تحت کنترل استفاده می شود.

7. روشهای مدل سازی فازی از جمله Takagi – Sugeno , Sugeno – Yasukawa را به دقت تشریح کنید و یک مثال با آنها حل کرده، مقایسه کنید.

مدل تاکاگی – ساگنو (TS) یک روش شناخته شده برای استنتاج فازی با کاربردهای فراوان در سیستم های فازی، کنترل فازی و کلا در مهندسی فازی می باشد. مزیت اصلی مدل TS این است که یک مدل ساده می باشد که می تواند در کاربردهای عملی بسیاری برای مدل کردن و کنترل سیستم های پیچیده به کار رود. بخاطر اهمیت آن، مدل TS توجه بسیاری از دانشمندان و مهندسانی را که در صنعت و تحقیقات عملی کار می کنند به خود جذب کرده است.

مدل TS ساده یا کلاسیک یا رایج شامل تالی های خطی و غیر فازی کننده مرکز ثقل می باشد همانگونه که توسط تاکاگی و ساگنو ارائه شد. اجازه دهید یک سیستم دینامیک دلخواه را با S نشان دهیم و یک سری عملیات در فضای سیستم تعریف کنیم.

تساوی: سیستم های S_1, S_2 با هم برابرند اگر و فقط اگر برای ورودی یکسان و شرایط اولیه یکسان، خروجی S_1 مساوی خروجی S_2 باشد. (for all $t \in (0, +\infty)$)

جمع: سیستم S، مجموع سیستم S_1, S_2 است اگر و فقط اگر برای ورودی یکسان و شرایط اولیه یکسان، خروجی S مجموع خروجی های S_1, S_2 باشد. (for all $t \in (0, +\infty)$)

ضرب در یک ثابت حقیقی: سیستم S حاصلضرب عدد حقیقی r و سیستم S_1 است اگر و فقط اگر برای ورودی یکسان و شرایط اولیه یکسان، خروجی S حاصلضرب عدد حقیقی r و خروجی S_1 باشد. (for all $t \in (0, +\infty)$)

ضرب: سیستم S حاصلضرب S_1, S_2 است اگر و فقط اگر برای ورودی یکسان و شرایط اولیه یکسان، خروجی S، حاصلضرب خروجی های S_1, S_2 باشد. (for all $t \in (0, +\infty)$)

سیستم پایدار: سیستم S پایدار است اگر برای هر ورودی محدود (تابع محدود در زمان)، خروجی هم محدود باشد (تابع محدود در زمان).

اجازه بدهید U مجموعه مرجع همه سیستم های دینامیک کلی با ورودی های محدود (محدود شده با یک ثابت مثبت M) و شرایط اولیه صفر باشد. در U می توانیم یک تابع از سیستم S را برای همه ورودی های محدود و شرایط اولیه صفر بصورت $\|S\| = \sup(\|y\|_n)$ تعریف کنیم. واضح است که این تابع یک نرم است چون:

$$\triangleright \|S\| > 0 \text{ است اگر } S \neq 0 \text{ و } \|S\| = 0 \text{ است اگر } S = 0.$$

$$\triangleright \|\lambda S\| = |\lambda| \cdot \|S\| \text{ برای هر عدد حقیقی } \lambda.$$

$$\triangleright \|S_1 + S_2\| \leq \|S_1\| + \|S_2\|$$

بنابراین U یک فضای نرم است. از طرفی دیگر، U یک فضای برداری هم می باشد که با توجه به تعاریف ذکر شده برای جمع و ضرب در یک عدد حقیقی می توان درستی آنرا نشان داد. لذا U یک فضای برداری متریک است. حال دنباله سیستم های S_1, S_2, \dots, S_k را با ویژگی خاص زیر در نظر بگیرید:

هر $\varepsilon > 0$ ، وجود دارد v که $d_n(S_1, S_m) < \varepsilon$ برای هر $m \geq v$. می توانیم اثبات کنیم که سیستمی مثل S وجود دارد که حد این دنباله می باشد. سیستم ها بصورت سیستمی تعریف می شوند که خروجی آن (برای ورودی ها محدود و شرایط اولیه صفر)، حد خروجی ها می باشد. این امر اثبات می کند که U یک فضای برداری نرم کامل است یعنی U یک فضای باناچ (Banach) است.

عبارت منطقی یا جمله منطقی، هر عبارت یا رابطه ای است که به اندازه μ صحیح می باشد $(\mu \in [0, 1])$. با این تفاسیل، مدل تاکاگی – ساگنو ساده یا استاندارد اینگونه بیان می گردد:

فرض کنید n قانون فازی مجزا داریم (یعنی از هم مستقل می باشند): If LE_i then Si , $i = 1, 2, \dots, n$

یعنی برای هر قانون اگر LE_i برقرار باشد، آنگاه y را از خروجی سیستم S_i با ورودی معین و شرایط اولیه معین محاسبه کن.

در حالتیکه بطور همزمان عبارات منطقی LE_1, LE_2, \dots, LE_n را با درستی $\mu_1, \mu_2, \dots, \mu_n$ داریم آنگاه کمیت y از روی خروجی سیستم S محاسبه می شود که :

$$S = \frac{\sum_{i=1}^N \mu_i S_i}{\sum_{i=1}^N \mu_i}$$

تعمیم مدل تاکاگی – ساگنو: فرض کنید n قانون فازی مجزا را داریم : LE_i then $S_i, i = 1, 2, \dots, n$. در این جا y از روی خروجی سیستم S محاسبه می گردد که

$$S = \frac{\sum_{i=1}^N \mu_i^p S_i}{\sum_{i=1}^N \mu_i^p}$$

که p یک ثابت صحیح مثبت می باشد. برای $p = 1$ مدل ساده TS را داریم. برای $p \rightarrow \infty$ حالت زیر را بدست می آوریم : $S = S_{\mu'}$ که $\mu' = \max(\mu_1, \mu_2, \dots, \mu_n)$. توسط بین ثابت شده است که سیستم های TS تعمیم یافته تقریب زننده های عام در حالت استون – وایرشراس می باشند. یعنی هر تابع پیوسته تعریف شده روی بازه $[a, b]$ را می توان بطور یکنواخت و با هر دقتی که می خواهیم با کمیت y یک سیستم TS تعمیم یافته تقریب بزنیم.

مدل تاکاگی – ساگنو مدل يك کنترل کننده غیرخطی است که از درونیابی کنترل کننده های خطی به دست می آید. سوگنو اثبات کرده است که فقط خروجی های Singleton برای يك کنترل کننده کفایت می کند. این مقدار Singleton تابعی از ورودی ها می باشد. صورت کلی يك قانون T- S^1 به صورت زیر می باشد:

IF $f(e_1 \text{ is } A_1, e_2 \text{ is } A_2, \dots, e_k \text{ is } A_k)$ THEN $y = g(e_1, e_2, \dots, e_k)$

¹ Takagi-Sugeno

مثلاً يك مدل از مرتبه صفر T-S به صورت زیر است:

IF error is zero AND change in error is zero THEN output = c

یا قوانین مرتبه اول T-S به این ترتیب می باشد:

*IF error is zero AND change in error is zero THEN output = a * error + b * change in error + c*

در این مدل استنتاج بین يك Rule Base که شامل چندین قانون می باشد، مشابه قبل است. هر قانون با درجه اعتقادی فعال می شود و نتیجه نهایی، غیر فازی شده نتایج به روش میانگین وزن دار است:

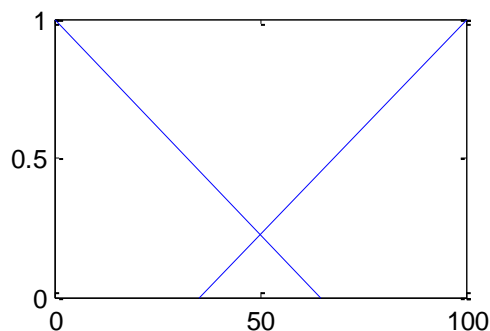
$$Z = \frac{\sum \alpha_i * output_i}{\sum \alpha_i}$$

واضح ترین مثال در این زمینه، مثال زیر است:

1. if error is Large then Output is Line1
2. if error is Small then Output is Line2

که در آن $Line1 = 0.2 * error + 90$ و $Line2 = 0.6 * error + 20$.

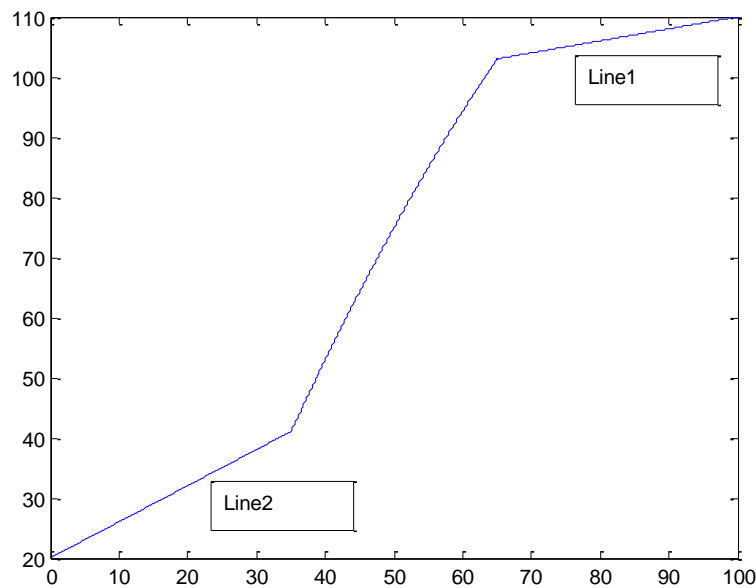
حال اگر توابع عضویت را به شکل ۷-۱ در نظر بگیریم:



شکل ۷-۱: توابع عضویت Small و Large

حال اگر توابع عضویت را هم مثلاً بصورت زیر در نظر بگیریم:

در نقاطی که دو تابع عضویت با هم تداخل دارند، خروجی به شکل ۷-۲ خواهد بود. (برای مشاهده این امر کافی است که از معادلات دو خط بالا درجه اعتقاد در هر نقطه ای در دامنه تداخل را به مجموعه های small و large درآورده و يك میانگین وزن دار با توجه به این مقدارهای وزن دار از دو معادله خط بگیریم). شکل ۷-۲ این غیرخطی بودن را نمایش می دهد.



شکل ۷-۲ : خروجی کنترل کننده T-S

مدل Sugeno-yasukawa

عصاره این روش استفاده از کلاسترینگ و مدل کردن فازی می باشد. در آوردن کلاسترها در فضای خروجی، به دست آوردن کلاسترها در فضای ورودی های مرتبط با هر خروجی و در نهایت استخراج قوانین، جمله ای است که می تواند بیان کننده این روش باشد.

در این روش ابتدا ورودی هایی را که فکر می کنیم در ساخت کنترل کننده سیستم دخیل هستند، مشخص می کنیم. معمولاً این کار سختی نیست و می توان این متغیرها را به دست آورد. سپس با مشاهداتی که انجام می دهیم، یک سری داده های ورودی یا خروجی های متناظر به دست می آید که در نقش Train set ما هستند.

مرحله بعدی مشخص کردن تعداد قوانین می باشد. این کار با استفاده از کلاسترینگ در فضای خروجی بر اساس تمامی داده های ورودی موجود انجام می پذیرد. روش کلاسترینگ مورد استفاده C-mean می باشد. تعداد کلاسترهای به دست آمده در این مرحله تعداد قوانین Rule Base ما را تعیین می کند.

حال برای هر قانون يك پارتیشن بندی روی فضای ورودی های مرتبط با آن قانون صورت می پذیرد تا ورودی های تاثیرپذیر در این قانون مشخص گردند. یعنی در این مرحله، ورودی ها، خروجی ها و تعداد قوانین مشخص می شود. تنها مساله ای که باقی می ماند، ساخت توابع عضویت است. زیرا، ممکن است کلاسترهای نتیجه شده به صورت يك مقدار زبانی نباشد، مثلاً شرط تحذب رعایت نشده باشد.

نکته دیگر این است که تعداد قوانین نهایی و تعداد کلاسترها روی خروجی الزاماً یکسان نیستند. چون امکان دارد هنگام کلاستر بندی روی ورودی های مربوط به يك قانون، فضای ورودی ها به بیش از يك قانون مربوط شود که به تعداد کلاسترهای اضافی (در اثر این رویداد)، تعداد قوانین افزایش می یابد. ورودی های مرتبط با قوانین از روی داده های موجود در هر کلاستر مشخص شده و کلاسترهای مختلف، قوانین مختلف را تعیین می کند.

کارایی مدل را می توان با محاسبه خطای میانگین مربعات روی مجموعه داده های موجود و آزمایش مدل روی آنها و محاسبه خطای موجود روی خروجی های دو سری داده واقعی و آزمایش شده، به دست آورد. در ادامه، يك سری اصلاحات روی متغیرها اعمال کرد و نتیجه را دوباره با تست خطای موجود ارزیابی کرد. به این صورت مدل به سمت بهینه شدن پیش می رود. برای ساخت توابع عضویت، ابتدا شکل آنها مشخص می شود. برای مثال تابع مثلثی، سپس، برای هر متغیر در قوانین، مقدار داده هایی که می تواند در کلاستر مربوطه بگیرد، روی محور رسم می شود. در ادامه با روشهای محاسبات عددی بهترین سه نقطه مشخص می شود، به طوری که فاصله سایر نقطه ها از آن مثلث کمینه باشد.

مدل T-S² پویا

این مدل مجموعه‌ای از قواعد ”اگر... آنگاه” است که مجموعه‌های فازی، بخش ”مقدم” آن و سیستمهای پویای LTI بخش ”تالی” آن را تشکیل می‌دهد. یک مدل T-S عام به شکل زیر است:

i^{th} Plant Rule: IF $x_1(t)$ is \bar{M}_{i1} and $\dots, x_n(t)$ is \bar{M}_{in} THEN $\dot{x} = A_i x + B_i u$

$$x \in R^{n \times 1}, i = \{1, \dots, r\}, A_i \in R^{n \times n}, B_i \in R^{n \times m}, u \in R^{m \times 1}$$

r تعداد قواعد است و x بردار حالت و \bar{M}_{ij} مجموعه‌های فازی ورودی هستند.

برای یک کنترل کننده فازی که روش استنتاج آن max-dot و غیرفازی ساز آن، میانگین مرکز است، می‌توان مدل بالا را به صورت زیر نوشت:

$$\dot{x} = \frac{\sum_{i=1}^r w_i(x)(A_i x + B_i u)}{\sum_{i=1}^r w_i(x)}$$

$$w_i(x) = \prod_{j=1}^n \mu_{ij}(x_j)$$

u_{ij} تابع عضویت مجموعه فازی i -ام در قاعده j -ام است. اگر α_i را به گونه زیر تعریف می‌کنیم:

$$\alpha_i(x) = \frac{w_i(x)}{\sum_{i=1}^r w_i(x)}$$

می‌توان معادله کنترل کننده را به شکل زیر بازسازی کرد:

$$\dot{x} = \sum_{i=1}^r \alpha_i(x)(A_i x + B_i u)$$

با به کارگیری همین روش برای ساختن کنترل کننده خواهیم داشت:

i^{th} Controller Rule: IF $x_1(t)$ is M_{i1} and $\dots x_n(t)$ is M_{in} THEN $u = -K_i x$

فرمول کلی کنترل عبارتست از:

$$u = - \sum_{i=1}^r \alpha_i(x) K_i x$$

که با جایگذاری مقدار u در عبارت اصلی، معادله نهایی به دست می آید:

$$\dot{x} = \sum_{i=1}^r \sum_{j=1}^r \alpha_i(x) \alpha_j(x) (A_i - B_i K_j) x$$

مثال:

مساله تراز کردن پاندول را روی يك ارايه مد نظر است. معادله حرکت پاندول به صورت زیر است:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{g \sin(x_1) - a m l x_2^2 \sin(2x_1)/2 - a \cos(x_1) u}{4l/3 - a m l \cos^2(x_1)} \end{aligned}$$

x_1 : زاویه پاندول نسبت به خط عمود

x_2 : سرعت زاویه پاندول

g : شتاب جاذبه 9.8 m/s^2

m : جرم پاندول 2 Kg

M : جرم ارايه 8 Kg

$2l$: طول پاندول 1.0 m

u : نیروی وارد شده به ارايه

(فرض شده: $a = \frac{1}{m+M}$)

باری مدل سازی فازی مساله، ۲ قاعده (Rule) طراحی شده است:

Plant Rule (1): If x_1 is close to zero Then

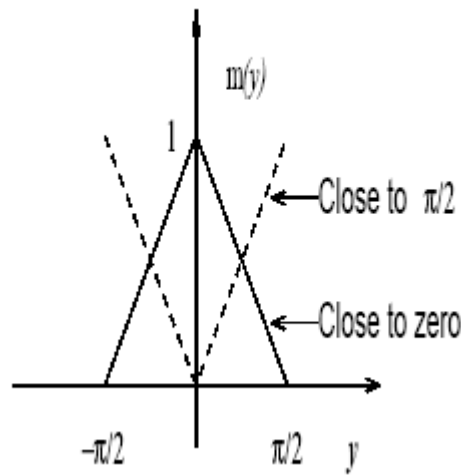
$$\dot{x} = A_1 x + B_1 u$$

Plant Rule (2): If x_1 is close to $\pm\pi/2$ Then

$$\dot{x} = A_2 x + B_2 u$$

۲ تابع عضویت u_1 و u_2 برای متغیرهای زبانی "نزدیک به صفر" و "نزدیک به $\pi/2$ " در

شکل ۷-۳ مشاهده می شود.



شکل ۷-۳: توابع عضویت "نزدیک به صفر" و "نزدیک به $\pi/2$ "

چون کنترل حول نقطه ۹۰ درجه امکان ندارد، سیستم حول نقطه ۸۰ درجه خطی می‌شود. مقادیر A_1 و A_2 و B_1 و B_2 ، مطابق زیر است:

$$\begin{aligned}
 A_1 &= \begin{bmatrix} 0 & 1 \\ \frac{g}{4l/3-aml} & 0 \end{bmatrix} & B_1 &= \begin{bmatrix} 0 \\ -\frac{a}{4l/3-aml} \end{bmatrix} \\
 A_2 &= \begin{bmatrix} 0 & 1 \\ \frac{9g}{4\pi(4l/3-aml\beta^2)} & 0 \end{bmatrix} & B_2 &= \begin{bmatrix} 0 \\ -\frac{a\beta}{4l/3-aml\beta^2} \end{bmatrix} \\
 \beta &= \cos(80^\circ)
 \end{aligned}$$

با استفاده از فرمولهای گفته شده می‌توان عملیات کنترل را انجام داد.

مدلسازی فازی مبتنی بر شبکه عصبی³

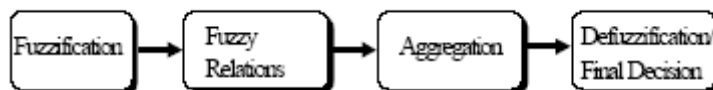
این قسمت نگاهی گذرا به مدلسازی فازی مبتنی بر شبکه عصبی دارد. با يك مثال از ابتدا تا انتها کلیات این مدلسازی بیان می‌شود. بنا به فرض، يك سیستم خبره آموزشی وجود دارد که هدف آن ارزیابی دانش‌آموزان در يك زمینه خاص است. این سیستم می‌خواهد نقش يك آموزگار را ایفا کند و با مدلسازی رفتار دانش‌آموزان نحوه پیشرفت آنها را در قالب اعداد و ارقام بیان کند. در ابتدا توضیحی کلی در مورد سیستم ارائه می‌شود.

³ Neural network-base fuzzy modeling

سیستم خبره می‌کوشد با ارزیابی از دانش‌آموزان، رفتار آنها را مدل کند. تنها راه تعامل دانش-آموزان با سیستم از طریق موس و صفحه کلید است. از این طریق دانش‌آموزان به سوالات سیستم پاسخ داده و سیستم پاسخ آنها را مورد ارزیابی قرار می‌دهد. تعداد پاسخ‌های صحیح و ناصحیح به سوالات، سطح دانش دانش‌آموزان را مورد ارزیابی قرار می‌دهد. گروه خاصی از سوالات، تصور غلط دانش‌آموزان را از مفاهیم ارزیابی می‌کند. زمان پاسخ‌گویی به سوالات و تعداد تلاش‌ها برای پاسخ دادن، از دیگر پارامترهایی هستند که به عنوان ورودی به سیستم وارد می‌شوند. در نهایت، کارایی یادگیری از طریق پارامترهایی همچون سطح دانش افراد، اشتباهات، تصور غلط از مفهیم، سرعت یادگیری، محدودیت حافظه و ... بیان می‌شود.

مدل فازی شبکه-عصبی

تمام ورودیها و متغیرهای بالا، متغیرهای زبانی هستند که با گذر از ۴ مرحله فازی سازی، سیستم روابط فازی، شبکه جمع‌آوری فازی و غیر فازی ساز پردازش می‌شوند. تمام این مراحل شبکه‌های پیوندگرا هستند. نمای کلی این سیستم در شکل ۷-۴ نمایش داده شده است. در ادامه به بررسی هر کدام از مراحل می‌پردازیم:

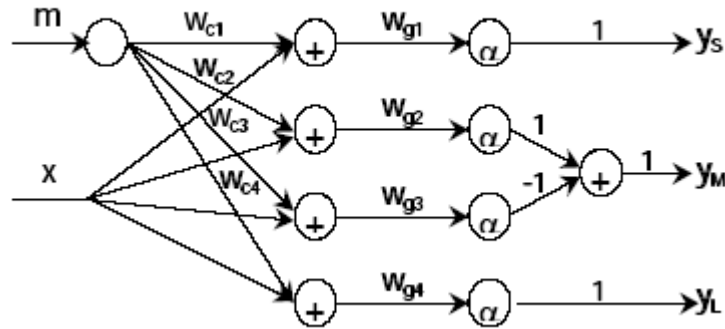


شکل ۷-۴: مراحل مدل فازی مبتنی بر شبکه عصبی

فازی سازی

ورودی‌های این سیستم يك سری اعداد و ارقام هستند که سیستم همانند يك آموزگار می‌باید در قالب مفاهیم زبانی با آنها برخورد کند. برای این منظور فازی سازی لازم است. در این مرحله، درجه اعتقاد به مفاهیمی همچون زمان پاسخ‌گویی، تعداد پاسخ‌های صحیح، تعداد تلاش‌ها و ... با توجه به مقادیر آنها، در قالب مفاهیم زبانی "کم"، "متوسط" و "زیاد" بیان می‌شود. شکل توابع

عضویت منظم است و محاسبه مقادیر با استفاده از شبکه عصبی با وزن ثابت⁴ انجام می‌شود. شکل ۵ شبکه عصبی مورد استفاده در این مرحله را نمایش می‌دهد.



شکل ۷-۵: شبکه عصبی مرحله فازی سازی

سیستم روابط فازی

خروجی مرحله فازی ساز باید به گونه‌ای به فضای خصوصیات دانش‌آموزان (متغیرهای خروجی نهایی) منتقل شود. این کار توسط سیستم روابط فازی انجام می‌شود. طی این مرحله، با کمک روابط فازی، متغیرهای ورودی سه بعدی (کم، متوسط و زیاد) به فضای متغیرهای خروجی پنج حالتی منتقل می‌شود. برای مثال، سرعت یادگیری دارای مفاهیم زبانی "کند"، "نسبتاً کند"، "عادی"، "نسبتاً سریع" و "سریع" است. روش استنتاج استفاده شده در این مرحله، max-min است. پیاده سازی این مرحله با چندین شبکه يك لایه با سه ورودی و پنج خروجی انجام می‌شود.

شبکه جمع‌آوری فازی

در خروجی مرحله قبل، با توجه به نوع اطلاعات، چندین مجموعه فازی برای يك مفهوم تولید می‌شود. وظیفه این مرحله به دست آوردن مجموعه فازی نهایی است. عملگر اجتماع برای این منظور استفاده می‌شود، زیرا، اجازه می‌دهد تمام خروجی‌های فازی در تشکیل مجموعه فازی نهایی شرکت کنند.

⁴ Fixed weight neural network

غیر فازی ساز

وظیفه این بخش، تولید مشخصه غیر فازی از خصوصیات دانش‌آموزان است. خصوصیات دانش‌آموزان در مرحله قبل در قالب يك مجموعه فازی محاسبه شده، به این بخش وارد می‌شود. غیر فازی ساز با کمک شبکه عصبی پس انتشار⁵، ارزیابی آموزگار را تقلید می‌کند. یادگیری این شبکه با الگوریتم BPVS است.

⁵ Back propagation

8. روشهای استخراج توابع عضویت فازی را به دقت تشریح نمایید.

توابع عضویت نقش محوری در تئوری مجموعه های فازی ایفا می نمایند. در کاربرد مجموعه های فازی بایستی توابع عضویت بطریقی ساخته شوند که بخوبی گویای معانی واژه های زبانی مربوطه باشند. البته این معانی بستگی تام به زمینه کاربردی آن دارد. مثلا واژه جوان هنگامیکه برای کودکان، اساتید دانشگاه و افراد بازنشسته بکار می رود، دارای معانی مختلفی است. معانی جوان حتی هنگامیکه برای اشیا مختلف مانند دورانه های زمین شناسی، درختان، ستارگان و غیره به کار می رود، متفاوت اند.

ساختن توابع عضویت در زمینه های گوناگون فی نفسه، مسأله اولیه تئوری مجموعه های فازی نیست. مسأله اولیه مربوط به اکتساب دانش بوده که در مهندسی دانش بدان پرداخته می شود. اکتساب دانش بوسیله مهندسین دانش و با همکاری متخصصین (Experts) حوزه های خاص انجام می گیرد. در اینجا نقش اصلی بوسیله مهندسین دانش که استخراج دانش مورد نیاز از متخصصین و مدارک مربوطه را انجام داده و آنرا به شکل عملی مورد نیاز در می آورند، ایفا می گردد. تاکنون روشهای گوناگونی برای ساختن توابع عضویت فازی ابداع شده است.

در مواردی منطقی است که از خود متخصص خواسته شود که تابع عضویت را برای واژه زبانی خاص بطور کامل یا بصورت نمونه برای بعضی از عناصر موجود در مجموعه مرجع تعریف نماید. درخواست برای تعریف کامل که غالبا بصورت یک فرمول ریاضی بیان می گردد، معمولا به نتیجه درست و قابل قبول منجر نمی گردد. این روش تنها برای واژه های زبانی که بصورت کامل در بعضی از مجموعه های مرجع وجود داشته و به نمونه ایده آل معروف بوده و انطباق سایر اعضا آن مجموعه با نمونه ایده آل بوسیله یک تابع تشابه انجام گرفته بکار برده می شود.

اگر تعیین تابع عضویت بصورت کامل مقدور نباشد، متخصصین اغلب قادر به بیان آن از طریق ارائه نمونه و مثال برای بعضی از عناصر انتخابی در مجموعه مرجع می باشند. تمثیل

و نمونه سازی را می توان با سوال کردن از متخصص درباره تطابق این عناصر نمونه که با متغیر x نشان داده می شود با واژه زبانی که بایستی در مجموعه فازی A قرار داشته باشد، انجام داد. پاسخ به سوالات بصورت مجموعه ای از زوج های $\langle x, A(x) \rangle$ درآورده می شود. این مجموعه برای ساختن تابع عضویت کامل مورد استفاده قرار می گیرد. روش دیگر برای ترسیم توابع عضویت، انتخاب توابع عضویت شناخته شده (مثلی، دوزنقه ای، زنگوله ای و غیره) و استفاده از روش تطابق منحنی (Curve Fitting) برای تعیین بهترین انطباق با نمونه مورد نظر است. این روش بسیار موفق بوده است. هم اکنون شبکه های عصبی بعنوان ابزاری مناسب برای ساختن توابع عضویت انطباقی بکار برده می شوند.

در مواردی که از چندین متخصص برای تعیین توابع عضویت استفاده می شود، بایستی عقاید و نظرات آنها را بطریق مناسب جمع بندی (Aggregate) نمود. بعنوان مثال فرض کنید پنج شناگر به نام های آلیس، بونی، کتی، دینا و اوا وجود داشته باشند. همچنین ده داور برای ارزیابی شناگران که با حرف r_1, r_2, \dots, r_{10} نشان داده شده اند، حضور دارند. ما باید تابع عضویت A که بیان کننده واژه زبانی "شناگر عالی" می باشد را تعیین کنیم. در اینجا از هر داور سوال می شود که نظر خود را درباره شناگران بالا که آیا کدامیک شناگر ماهرند را ارائه نمایند. پاسخ باید بصورت بله یا خیر باشد. فرض کنید جدول زیر نتیجه پاسخ داوران بوده و اعداد یک و صفر بجای بله یا خیر بکار رفته باشند. سپس برای هر شناگر درجه عضویت در مجموعه A با قرار دادن نسبت جمع اعداد یک به تعداد کل داوران محاسبه می گردد. حاصل این عملیات، مجموعه فازی زیر خواهد بود:

$$A = 0.3/\text{Alice} + 0.4/\text{Bonnie} + 0.6/\text{Cathy} + 0.9/\text{Dina} + 0.6/\text{Eva}$$

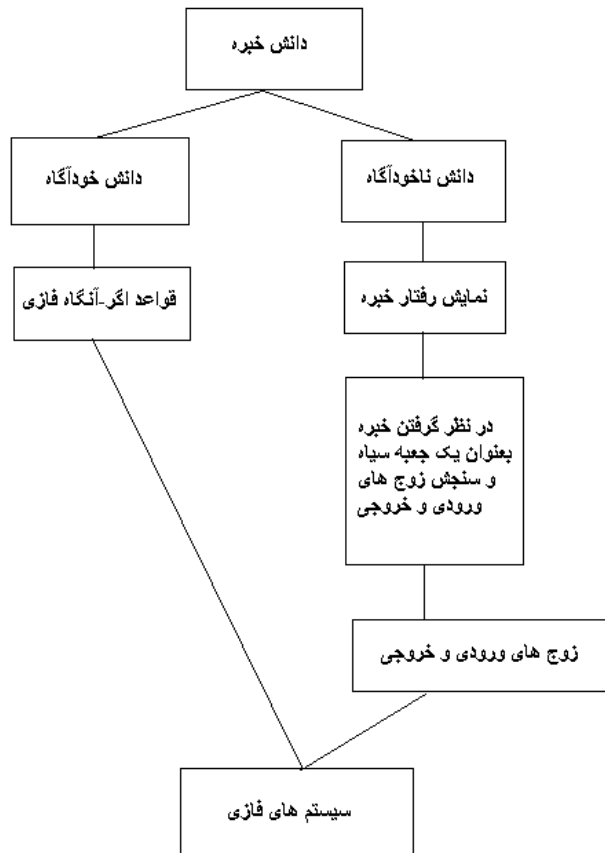
Eva	Dina	Cathy	Bonnie	Alice	
1	1	1	1	1	r_1
1	1	1	0	0	r_2
0	1	0	1	0	r_3
1	1	1	0	1	r_4
1	1	1	0	0	r_5

1	1	1	1	0	r ₆
0	0	0	0	0	r ₇
1	1	1	1	1	r ₈
0	1	0	0	0	r ₉
0	1	0	0	0	r ₁₀

سیستم های فازی برای فرموله کردن دانش بشری استفاده می شوند. بنابراین یک سوال مهم این است که: دانش بشری معمولاً چه شکلی است؟ بصورت عامیانه دانش بشری را در زمینه یک مسأله فنی خاص می توان به دو دسته تقسیم کرد: دانش خودآگاه و دانش ناخودآگاه. در دانش خود آگاه، منظور ما این است که دانش را می توان بصورت صریح و روشن در قالب کلمات بیان کرد و در دانش ناخودآگاه منظور ما وضعیت هایی است که انسان های خبره می دانند چه کاری انجام می دهند ولی نمی توانند آنرا بطور دقیق در قالب کلمات بیان کنند. بعنوان مثال رانندگان باتجربه کامیون می دانند که در شرایط سخت چگونه رانندگی کنند ولی نمی توانند عملکردشان را در قالب کلمات بیان کنند(آنها دارای دانش ناخودآگاه هستند). این افراد حتی اگر بتوانند رفتارشان را در قالب کلمات بیان کنند، این توصیف معمولاً برای انجام آن کار ناکافی است. در دانش خودآگاه ما خیلی ساده می توانیم از انسان خبره بخواهیم رفتار خود را در قالب عبارتهای اگر – آنگاه فازی بیان کرده و آنرا در سیستم های فازی قرار دهیم.

در دانش ناخودآگاه، آنچه که می توانیم انجام دهیم این است که از انسان خبره بخواهیم که رفتار خود را نمایش دهد. بدین معنی که آنچه آنها در وضعیت خاص انجام می دهند را نشان دهند. هنگامی که خبره در حال نمایش است ما او را بدیده جعبه سیاه نگریسته و ورودی ها و خروجی های او را بسنجیم. یعنی مجموعه ای از داده های ورودی – خروجی را جمع آوری کنیم. در این حالت، دانش ناخودآگاه به مجموعه ای از زوج های ورودی – خروجی تبدیل می شود، بنابراین مسأله اساسی و مهم، ساخت سیستم های فازی از روی زوج های ورودی – خروجی است.

از روشهای ساخت سیستم های فازی می تواند به طراحی سیستم های فازی با استفاده از جدول جستجو، با استفاده از روش گرادیان نزولی، با استفاده از روش کوچکترین مربع های بازگشتی و با استفاده از خوشه سازی اشاره کرد.



9. روشهای مختلف فیلترینگ فازی که در بهینه سازی کیفیت تصاویر استفاده می شوند را مرور نمایید.

کاربرد تکنیک های فازی در پردازش تصویر، یک زمینه تحقیقاتی جالب توجه است. تکنیک های فازی در حوزه های گوناگونی از پردازش تصویر اعمال شده اند (مثلا درونیابی فازی و مورفولوژی) و کاربردهای عملی متعددی دارند (مثلا پردازش تصویر صنعتی یا طبی).

هم اکنون فیلترهای فازی گوناگونی برای کاهش نویز ایجاد شده اند مثلا FIRE-filter معروف یا فیلتر میانگین وزندار فازی و فیلتر تکراری مبتنی بر کنترل فازی. اکثر تکنیک های فازی در کاهش نویز تصویر، عمدتاً با نویزهای fat-tailed مانند نویزهای ضربه ای (impulse) سروکار دارند. این فیلترهای فازی قادر به انجام دادن روش های فیلترکردن مبتنی بر رتبه هستند (مانند فیلتر میانگین). به هر حال، اکثر تکنیک های فازی بطور خاص برای نویز (شبه) گاوسی طراحی نشده اند یا نتایج متقاعد کننده ای نمی دهند وقتی که برای کنترل این گونه نویز به کار می روند.

کاهش نویز در تصاویر، یکی از پایه ای ترین عملیات پردازش تصویر است. در سالهای اخیر، فیلترهای مبتنی بر منطق فازی، نشان داده اند که فیلترینگ موثری در تصویر فراهم می آورند. فیلترهای فازی می توانند بصورت های زیر باشند:

- فازی کردن فیلترهای میانگین وزندار.
- ترکیب فازی خروجی های زیر – فیلتر های کلاسیک متعدد.
- فیلترهای فازی خالص، که در آنها هیچ عملگر فیلتر کلاسیکی استفاده نشده است بلکه خروجی فیلتر، مستقیماً با یک مجموعه قوانین تعیین می شود.

فیلترهای ارائه شده توسط مانیا سو و تاگوچی، تاکاشیما و روسو به نوع اول فیلترها تعلق دارند. درحالیکه فیلترهای ارائه شده توسط چویی و کریشناپوران، تاگوچی و موگرو، مثال هایی از فیلترهای نوع دوم می باشند. FIRE filter یک مثال از یک فیلتر فازی خالص می

باشد. هیچ فیلتر عمومی و بهینه برای همه انواع و سطوح ممکن نویز وجود ندارد، ولی به جای آن فیلترها معمولاً برای یک نوع خاص از نویزها طراحی شده و روی نوع خاصی از نویزها تست می گردند. دو نوع عمده از نویزها، نویز گاوسی و نویز فلفل – نمکی می باشند.

حتی اگر ادعا شود که فیلتری طراحی شده است که با نویز ضربه ای سروکار دارد، ممکن است فقط روی فلفل – نمکی تست شود و در حقیقت بدتر عمل کند وقتی که دامنه ضربه های نویز تصادفی باشد. فیلترها می توانند بنا بر معیارهایی طراحی شوند مثل اندازه خطای عینی (Objective Error Measure: MSE, SNR)، منظر بصری تصویر، زمان پردازش، قابلیت پیاده سازی سخت افزاری، قابلیت تنظیم شدن و استحکام و دوام در مقابل تغییرات در توزیع نویز.

فیلترهایی طراحی شده اند که نویزهای مخلوط را کاهش می دهند (گاوسی و ضربه ای) و قادر به عمل در زمان واقعی (real time) هستند وقتی که نرخ فریم بالاتر از نرخ فریم های معمولی ویدئو باشد. این فیلترها بر اساس ارزیابی شباهت های فازی بین پیکسل ها در یک پنجره پردازشی محلی بنا شده اند. خروجی یک مجموعه از زیر – فیلتر ها سپس ترکیب می شوند تا خروجی نهایی فیلتر را فراهم سازند.

می توانیم مراجع زیر را در مورد جزئیات روش های فیلترینگ فازی بیان کنیم:

- [1] Y. Choi and R. Krishnapuram, A robust approach to image enhancement based on fuzzy logic. IEEE Transactions on Image Processing, 6(6) (1997) 808–825.
- [2] Y-H Lee and S. A. Kassam, Generalized median filtering and related nonlinear filtering techniques, IEEE Trans. Acoustics, Speech and Signal Processing, ASSP-3(3) (1985) 672–683.
- [3] Y.Wada M. Muneyasu and T. Hinamoto, Edge-preserving smoothing by adaptive nonlinear filters based on fuzzy control laws, Proc. of IEEE Int. Conference on Image Processing, ICIP'96, Lausanne, Switzerland (1996) 785–788.

[4] F. Russo, Noise cancellation using nonlinear fuzzy filters, Proc. of IEEE Instrumentation and Measurement Technology Conference IMTC'97, Ottawa, Canada, vol. 2 (1997) 772–777.

[5] A. Taguchi and M. Meguro, Adaptive L-filters based on fuzzy rules, Proc. of IEEE Symposium on Circuits and Systems, ISCAS'95, Seattle, WA (1995) 961–964.

[6] A. Taguchi, H. Takashima, and F. Russo, Data-dependent filtering using the fuzzy inference, Proc. of IEEE Instrumentation and Measurement Technology Conference, IMTC'95, Waltham, MA (1995) 752–756.

10. سیستم های ناوبری روباتیک را که در آنها از روشهای فازی استفاده شده مرور و تشریح نمایید.

اولین و معمولترین کاربرد تکنیک های منطق فازی در حوزه روبوتیک خودکار، استفاده از کنترل فازی برای پیاده سازی واحدهای رفتاری است. کنترلرهای منطق فازی، دانش کنترلی هیوریستیک را به شکل قوانین اگر – آنگاه اعمال می کنند و یک گزینه مناسب هستند وقتی که یک مدل خطی دقیق از سیستم تحت کنترل به آسانی بدست نمی آید. آنها همچنین درجه خوبی از استحکام در مواجهه با تغییر و عدم قطعیت در پارامترها از خود نشان داده اند. این ویژگی ها، کنترل فازی را مخصوصا برای ناوبری خودکار روبوت بسیار مناسب می سازد.

نوع کلاسیک کنترلرها یا رفتارهایی که در روبوت های متحرک استفاده شده بود، تعقیب مسیر می باشد. به کنترلر یک مسیر به صورت دنباله ای از مختصات در یک دستگاه مرجع داده می شود و آن دستورات موتور را تولید می کند تا مسیر را هر چه نزدیکتر و دقیقتر که می تواند تعقیب کند. برای انجام این کار، کنترلر باید موقعیت روبوت نسبت به مسیر و کینماتیک و دینامیک روبوت را بداند. موقعیت معمولا از اندازه گیری های کدکننده های چرخ یا حسگرهای داخلی دیگر بدست می آید: رفتار خود تحریک.

تعقیب مسیر می تواند بطور غیرمنتظره ای دشوار باشد. کینماتیک و دینامیک روبوت ممکن است پیچیده و غیر خطی باشد و تعامل بین وسیله و زمین برای مدل کردن دشوار باشد. این مسائل استفاده از کنترل فازی در تعقیب مسیر را ایجاب می کند. یک پیشنهاد توسط ایسیک داده شده است که تعقیب یک مسیر گسسته را در نظر می گیرد ولی نتایج عملی ارائه نمی دهد. بزرگوگ و ژانگ کنترلرهای فازی پیشنهاد داده اند برای مسأله مسیر – از بین – نقاط – راه. کار اخیر بر اساس شبیه سازی منحنی های اسپلاین می باشد.

محققین دیگر، کنترل فازی را روی یک نسخه ساده شده از تعقیب مسیر اعمال کرده اند: تعقیب یک خط مستقیم. نیشیموری یک سری از آزمایشات را گزارش می دهد که ترکیبات

مختلفی از عملگرهای فازی و تکنیک های غیرفازی سازی را امتحان کرده است. تاناکا و سانو قوانین فازی برای تعقیب خط را از یک مدل فازی وسیله کنترل شده استخراج می کنند. این کار به آنها این امکان را می دهد که یک اثبات پایداری را با توجه به تاناکا و سوگنو ارائه دهند. هر دو کار با شبیه سازی کامپیوتری تست شده اند.

نهایتا اولرو و همکارانش، دو راه برای اعمال کنترل فازی در تعقیب مسیرهای پیوسته دلخواه در نظر گرفته اند. در اولی، کنترلر فازی مستقیما کنترل روبات را تولید می کند. در دومی، پارامترهای استفاده شده توسط یک روش هندسی تولید می شوند. هر دو روش روی یک روبات واقعی با نتایج خوبی تست شده اند.

یک رفتار مبتنی بر حسگر، سیاست کنترلی را بر اساس حس کردن خارجی پیاده سازی می کند. روبات بیشتر با توجه به ویژگی های محیط حرکت می کند تا با توجه به مسیر ارائه شده داخلی. این نوع از کنترل بعنوان کنترل مطیع هم شناخته می شود. مثال های کلی می تواند رفتن در امتداد یک دیوار یا یک حد مرزی، رسیدن به یک منبع نور یا چراغ و اجتناب از موانع باشد.

اولین استفاده گزارش شده از کنترل فازی در روباتیک متحرک متعلق به این نوع است. مثلا در 1985، ساگنو و نیشیدا یک کنترلر فازی ایجاد کردند که قادر به راندن یک ماشین مدل در امتداد یک مسیر محدود به دو دیوار بود. یک حسگر صوتی ساده، برای اندازه گیری فاصله از دیوارها بکار رفته بود. کنترلر فازی روی یک ریزپردازنده 8080 اجرا می شد و از قوانین هیوریستیکی استفاده می کرد که از مطالعه رفتار یک راننده انسانی بدست می آمد. محققین نتایج تشویق کننده ای را گزارش دادند ولی کنترلر نسبت به خطاهای حسگرها استحکام نداشت و سرعت آن بسیار کم و محدود به 1.7 cm/sec بود.

مدت زمان کوتاهی بعد از ماشین ساگنو و نیشیدا، تاکچی یک کنترلر فازی برای اجتناب از مانع ارائه داد. کنترلر از یک الگوریتم ساده برای بدست آوردن اطلاعات درباره نواحی

خالی در جلوی روبات با یک دوربین ویدئو استفاده می کرد. قوانین بصورت تجربی و با کمک یک شبیه ساز بدست آمدند.

پین و بندر، نوع دیگری از رفتار را ارائه دادند که پردازش حافظه نامیده می شد و وظیفه آن مدیریت یک حالت داخلی بود تا چرخه های محدود را تشخیص دهد و از آنها فرار کند. روبات های بسیاری با فهرست رفتارهای متفاوتی مجهز شده اند که همه زیر – وظایفی را که باید انجام دهند در بر می گیرد. مثلا روبات های خودکار FLAKEY, MARGE, MORIA, LABOT. این روبات ها رفتارهای فازی برای رفتن به یک موقعیت خاص، آغاز حرکت به سمت یک مقصد، چسبیدن به یک شی، رد شدن از یک در و غیره را دارا می باشند.

کنترلرهای فازی معمولا طوری طراحی می شوند که یک هدف را در نظر بگیرند. اگر بخواهیم دو (یا چند) هدف متعامل را در نظر بگیریم، دو گزینه داریم. می توانیم یک مجموعه از قوانین پیچیده را بنویسیم که مقدم های آنها هر دو هدف را همزمان در نظر می گیرند؛ یا می توانیم دو مجموعه از قوانین ساده را بنویسیم که هر کدام مخصوص یک هدف باشد و خروجی آنها را به نحوی ترکیب کنیم. مثلا اگر دو هدف، تعقیب یک مسیر و اجتناب از مانع باشد، راهکار اول دارای قوانین فازی به شکل زیر می باشد:

اگر شرط – مسیر 1 و شرط – مانع 1 آنگاه دستور 1

اگر شرط – مسیر 2 و شرط – مانع 2 آنگاه دستور 2

در راهکار دوم، قوانین به شکل زیر هستند:

اگر شرط – مسیر آنگاه دستور 1

اگر شرط – مانع آنگاه دستور 2

راهکار اول برای ناوبری به سمت یک هدف همراه با اجتناب از موانع توسط اسکوبیک و همکاران استفاده شده است که از یک روبات مینیاتوری مبتنی بر مادون قرمز استفاده کردند؛ و توسط لی استفاده شده است که از یک روبات مبتنی بر صوت شبیه سازی شده استفاده کرد. این راهکار همچنین توسط آلتروک روی یک ماشین مدل بکار رفت برای تعقیب یک مسیر مسابقه محدود به دیوار همزمان با ممانعت از لغزش و انحراف به دلیل سرعت بالا.

راهکار دوم توسط ین و فلاوگر و باکستر و بامبی برای تعقیب مسیر همراه با اجتناب از موانع استفاده شد؛ و توسط مائدا برای تعقیب دیوار بصری همزمان با اجتناب از مانع روی یک روبات Hero2000 استفاده شد.

روبات های مبتنی بر رفتار فازی، کارایی عالی از خود در رویدادهای رباتیک عمومی نشان داده اند. تیم روبات های PIONEER از SRI International ، در مسابقات رباتیک AAAI 1996، رویداد "Call a Meeting" را برد. MORIA از GERMAN GMD جایزه هوش را در مسابقات روبات IEEE Fuzz'1995 دریافت کرد. MARGE از NCSU ، رویداد مرتب سازی دوباره دفترکار در مسابقات AAAI 1993 را برد؛ و FLAKEY از SRI International، در مسابقات AAAI 1992 در مکان دوم قرار گرفت. استفاده از منطق فازی اغلب منجر به حرکت بسیار روان و واکنش قابل اطمینان شده است. آنگونه که داور مربوطه درباره FLAKEY گفته است: "تنها رباتی که حس کردم می توانم در جلوی آن بنشینم یا دراز بکشم" (کاری که واقعا انجام داد!).

بیش از یک دهه منطق فازی در سیستم های ناوبری رباتهای متحرک بکار رفته است. در اوایل 1991 Yen , Pfluger یک روش برنامه ریزی مسیر برای کنترل رباتهای متحرک با استفاده از منطق فازی پیشنهاد دادند. از آنموقع کارایی بکارگیری منطق فازی در ناوبری رباتهای متحرک آشکار شده است. روشهای بسیاری برای ناوبری سیستم های رباتیک در محیط های ناشناخته بر اساس منطق فازی ارائه شده است. بر خلاف همه فوایدی که الگوریتم های فازی دارند هنوز جای پیشرفت بسیاری دارند. یکی از مشکلات موجود در ناوبری بر اساس

منطق فازی مسئله هدف غیر قابل دسترس⁶ است. در تحقیقاتی که در دانشگاه Western Ontario انجام شده است مشاهده گردیده که حداقل سه حالت وجود دارد که روبات قادر به رسیدن به هدف نیست.

- روبات برخی مواقع گم می شود هر چند که يك مسیر تا هدف وجود دارد
 - مسیر منتهی به مکان هدف بوسیله يك مانعی که یا بزرگ است و یا دارای يك سطح مقعر به سمت روبات است مسدود شده است.
 - مکان هدف دارای موانعی در همسایگی اش است.
- اولین مشکل ناشی از يك نقطه ضعف رایج سیستم های ناوبری واکنشی است. که این با بکارگیری يك طرح path-planning جامع حل می شود. مشکل دوم توسط الگوریتم های جدیدی برای حالاتی که روبات دارای رفتار سرگردان و یا نوسانی است قابل حل است.

الگوریتم ناوبری فازی

مسئله ناوبری روبات را می توان به دو زیر مسئله تقسیم کرد: یکی رسیدن به هدف و دیگری ممانعت از برخورد با وسایل اطراف. هدف اصلی يك الگوریتم منطق فازی استفاده از قوانین فازی کلامی برای جلب مفاهیم مبهمی که به وسیله مجموعه های فازی در سیستم های ناوبری ایجاد می شود است. استراتژی ناوبری فازی اثر دافعه ای⁷ که مربوط به فاصله و زاویه بین روبات و مانع است را با اثر جاذبه ای که به وسیله اختلاف زاویه بین جهت حرکت روبات و مکان هدف نهایی تولید می شود، ترکیب می کند. تعریف جهتی که در سیستم ناوبری روبات متحرك استفاده می شود معمولاً به فرمی است که در شکل 10-1 نشان داده شده است. فضای مرجع به صورت زیر تعریف شود

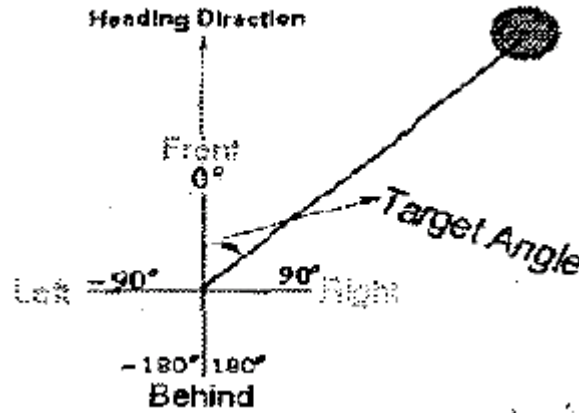
$$X = [-180^{\circ}, 180^{\circ}]$$

جهت حرکت رو به جلو روبات با صفر درجه نشان داده می شود و درجه های منفی برای سمت چپ و درجه های مثبت برای سمت راست در نظر گرفته

⁶ Goal-unreachable

⁷ repelling

می‌شود. برای راحتی محاسبات توابع عضویت در سیستم‌های ناوبری رباتهای متحرك به صورت مثلثی و یا دوزنقه‌ای در نظر گرفته می‌شود.



شکل 10-1 تعریف جهتها

مجموعه‌های فازی که به یک سیستم ناوبری ربات متحرك اعمال می‌شود به صورت زیر است:

1. جهت غیر مجاز: جهتی که احتمال برخورد با ربات وجود دارد و به صورت $\mu_{DisallowedDirection}$ نشان داده می‌شود.

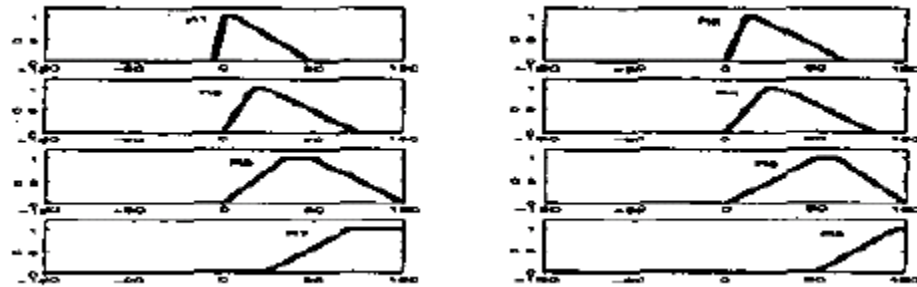
2. جهت مجاز: جهتی که برای ربات امن است و به صورت $\mu_{allowedDirection}$ نشان داده می‌شود.

3. جهت مورد انتظار: جهت مورد انتظار که ربات برای رسیدن به هدف بایستی تعقیب کند و به صورت $\mu_{DisiredDirection}$ نشان داده می‌شود.

4. جهت turning: جهتی که هم امن و هم مورد درخواست است و به صورت $\mu_{TurningDirection}$ نشان داده می‌شود.

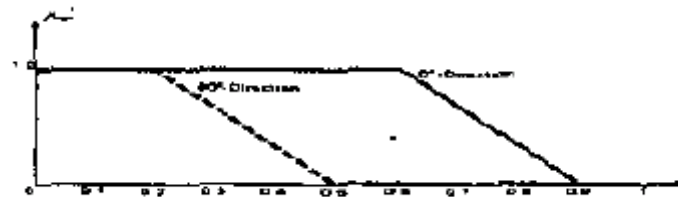
5. نزدیک: فاصله معمول (d) که بوسیله یک حسگر تشخیص داده می‌شود و به صورت μ_{near} نشان داده می‌شود.

هر حسگر بروی ربات از یک جهت مراقبت می‌کند که توسط یک مجموعه فازی با تابع عضویتش مشخص می‌شود. برای مثال در شکل 10-2 توابع عضویت ربات Koala را نشان می‌دهد.



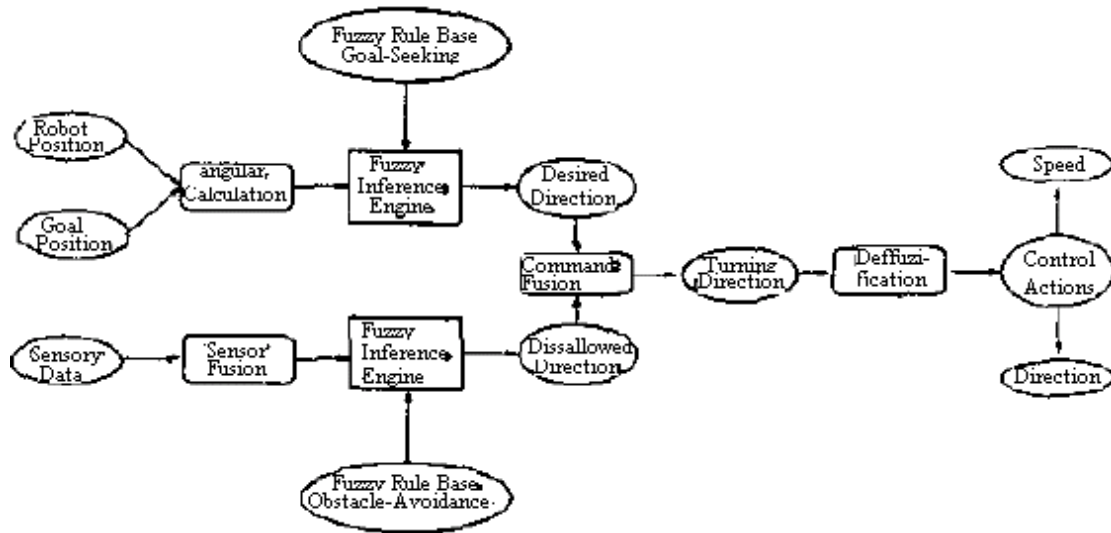
شکل 10-2 توابع عضویت جهات

در این شکل توابع عضویت سمت راست مربوط به حسگرهای سمت راست و توابع عضویت سمت چپ مربوط به حسگرهای سمت چپ است. از آنجائیکه يك مانع در جلوی روبات مهمتر از موانع در کنار و یا پشت سر هستند بنابراین هر جهت يك تابع عضویت يكتا Near دارد که به حسگر آن مربوط است. برای مثال در شکل 10-3 توابع عضویت مجموعه فازی Near حسگرهای صفر درجه و نود درجه در الگوریتم پیاده سازی شده در روبات Koala نشان داده شده است.



10-3 توابع عضویت Near

با توجه به این توابع عضویت يك مانع با اندازه 0.6 در حسگر صفر درجه دارای ماکسیمم درجه برای بکار افتادن قانون است و بدین معنی است که يك مانع در روبرو وجود دارد. در شکل 10-4 يك سیستم کنترل ناوبری روباتهای متحرك بر اساس منطق فازی را نشان می دهد



شکل 4-10 پیکر بندی سیستم کنترل ناوبری فازی

همانگونه که در شکل مشاهده می‌شود Fuzzy inference engine بر اساس قوانین فازی برای ممانعت از برخورد disallowed-direction ها را تولید می‌کند. که در واقع با گرفتن اطلاعات حسگرها به عنوان ورودی جهت‌های مجاز را به صورت محاسبه معکوس جهات غیر مجاز به دست می‌آورد که به صورت زیر است:

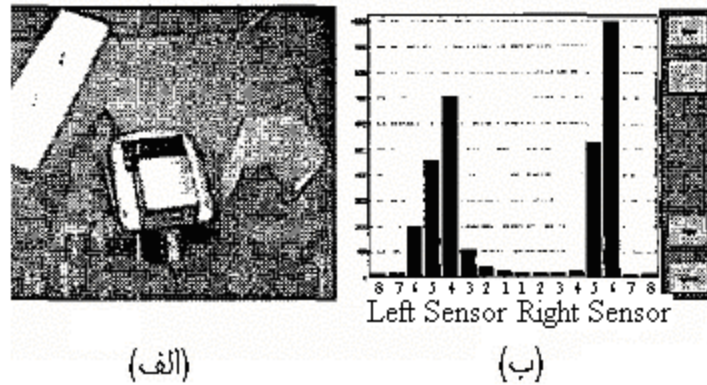
$$\mu_{Allowed-Direction} = 1 - \mu_{Disallowed-Direction}$$

همچنین Fuzzy inference Engine با استفاده از قوانین فازی برای Goal-seeking جهت مورد انتظار را تولید می‌کند و از آنجائیکه زاویه انحراف نهایی روبات بایستی هم شامل Desired (از قسمت ممانعت از برخورد) و هم شامل allowed باشد بنابراین Turning-Direction توسط اتصال این دو رفتار با استفاده از عملگر t-norm به صورت زیر محاسبه می‌شود:

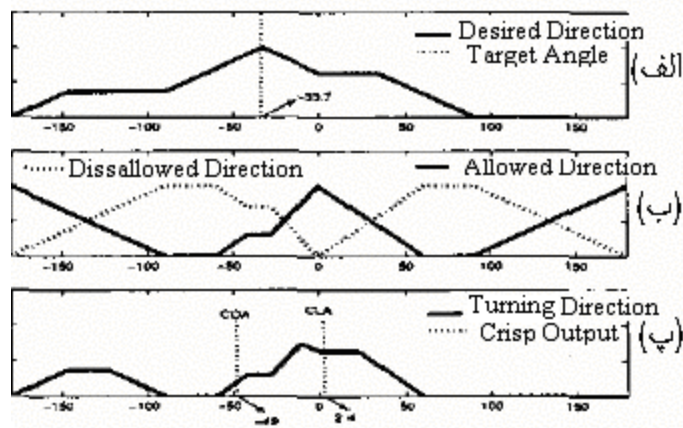
$$\mu_{Turning-Direction} = \min(\mu_{DesiredDirection}, \mu_{AllowedDirection})$$

جهت صریح نهایی به وسیله روش غیر فازی سازی CLA (Centroid of Largest Area) بدست می‌آید.

به عنوان يك نمونه ، روبات Koala در يك لحظه مشخص در شكل 10-5 الف نشان داده شده است . زاویه هدف 33.7- محاسبه شده است و نتایج خوانده شده از 16 حسگر در شكل 10-5 ب نشان داده شده است.



شكل 10-5 . (الف) koala در يك لحظه مشخص (ب) اطلاعات خوانده شده از 16 حسگر حسگرها برای بدست آوردن فاصله بین روبات و مانع تنظیم گردیده اند. هر چه عدد خوانده شده بزرگتر باشد به معنی نزدیکتر بودن به هدف است. ماکسیمم عدد خوانده شده 1023 است که برابر با فاصله 0-5cm از مانع است. در شكل 10-6 يك نمونه از خروجی کنترلر ناوبری فازی را در این لحظه نشان می دهد.

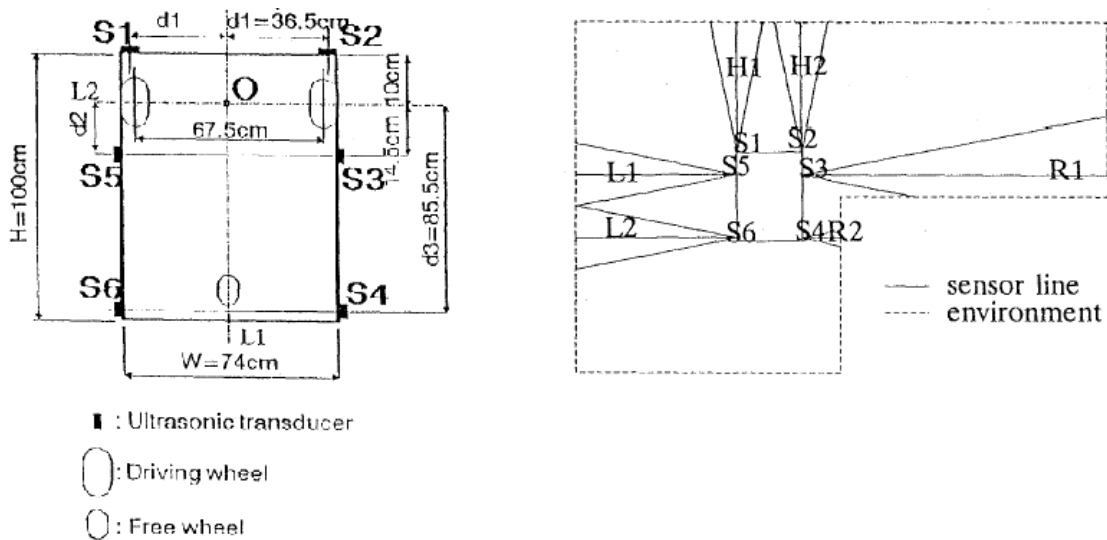


شكل 10-6 خروجی کنترلر ناوبری فازی در لحظه نشان داده شده در شكل 10-5

زاویه هدف برای تولید مجموعه فازی Desired-Direction همانگونه که در شكل 10-6 الف ملاحظه می گردد استفاده می شود. بر اساس

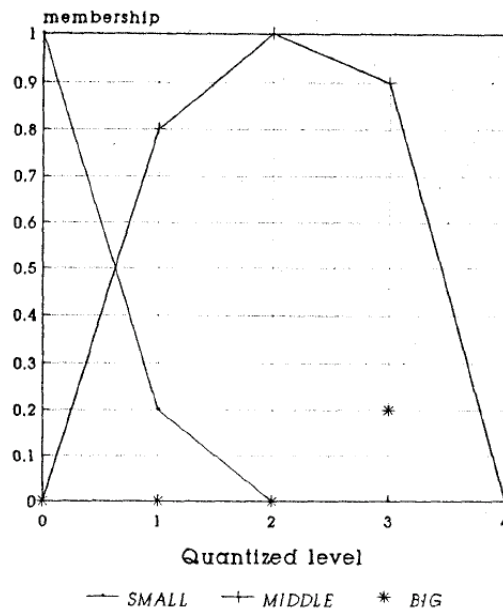
داده‌های حسگرها ملاحظه می‌شود که حسگرهای L4, L5, L6, R5, R6 داده‌هایی با مقدار بیشتر از باقی حسگرها دارند. با به هم پیوستن با تابع عضویت Near، کنترلر جهت‌های مجاز و غیر مجاز را همانگونه که در شکل 6-10 ب نشان داده شده است تولید می‌کند. Turning-Direction با استفاده از عملگر min بروی allowed-direction و disallowed direction همانند شکل 6-10 پ بدست می‌آید و سپس غیر فازی سازی CLA نقطه 2.4 درجه را در خروجی تولید می‌کند.

از دیگر موارد استفاده از سیستم ناوبری، در کنترل روبات متحرک در محیط ناشناخته است که يك نمونه از آن توسط K.Tai و J.Chau انجام گرفته است. در این روبات از دو چرخ رانشگر استفاده شده است. اگر سرعت چرخ راست بزرگتر از سرعت چرخ چپ باشد به سمت چپ می‌پیچد و برعکس. برای حرکت در مسیر مستقیم بایستی سرعت چرخ چپ و راست برابر باشد. روبات به شش حسگر التراسونیک مجهز شده است. شکل مکانیکی روبات و محل قرار گرفتن حسگرها در شکل 7-10 نشان داده شده است.



شکل 7-10 شکل مکانیکی روبات و مقادیری که حسگرها اندازه‌گیری می‌کنند

به علت اینکه تحلیلی اطلاعات شش حسگر به علت تعداد قوانین زیاد مشکل است آنها را به دو دسته $(s1, s2, s3, s4)$ برای چرخش به راست و $(s1, s2, s5, s6)$ برای چرخش به چپ تقسیم می‌کنیم و دو کنترلر فازی طراحی می‌شود. به منظور کاهش پیچیدگی از سه حالت کلامی (SMALL, MIDDLE, BIG) استفاده می‌شود هر حالت کلامی 5 سطح کوانتیزه شده دارد که عبارت است از $(0, 1, 2, 3, 4)$. تابع عضویت در شکل 8-10 نشان داده شده است.



شکل 8-10 تابع عضویت برای مجموعه فازی

مجموعه $(S1, S2, S3, S4)$ فاصله های $(H1, H2, R1, R2)$ را تشخیص می‌دهند. سه مجموعه فازی (SMALL, MIDDLE, BIG) و چهار متغیر برای ساخت 81 قانون استفاده می‌شوند. با استفاده از این قوانین روبات متحرك بدون برخورد با موانع می‌تواند به سمت راست بچرخد. به این کنترلر RFLC (RIGHT FUZZY LOGIC CONTROLLER) گویند. جدول تصمیم گیری دارای شامل 625 ترکیب است. به علت اینکه فاصله بین 30 تا 100 سانتیمتر است در ابتدا کوانتیزه می‌شود و به دو گروه به صورت

شکل 9-10 تقسیم می‌شوند. پس از عمل کوانتیزاسیون خروجی حسگرها به صورت (LH1,LH2,LR1,LR2,LL1,LL2) بیان می‌شود.

Group 1: (H1,H2)

H1 and H2	Quantized level of H1 and H2
0 – 69cm	0
70 – 129cm	1
130 – 139cm	2
140 – 169cm	3
169 – 1000cm	4

Group 2: (L1,L2,R1,R2)

R1,R2,L1,L2	Quantized level of R1,R2,L1,L2
0 – 49cm	0
50 – 67cm	1
68 – 71cm	2
72 – 92cm	3
93 – 1000cm	4

شکل 9-10 گروه‌بندی داده‌ها

سپس با استفاده از مجموعه قوانین همانند قوانین زیر جدول تصمیم‌گیری به شکل 10-11 تولید می‌شود:

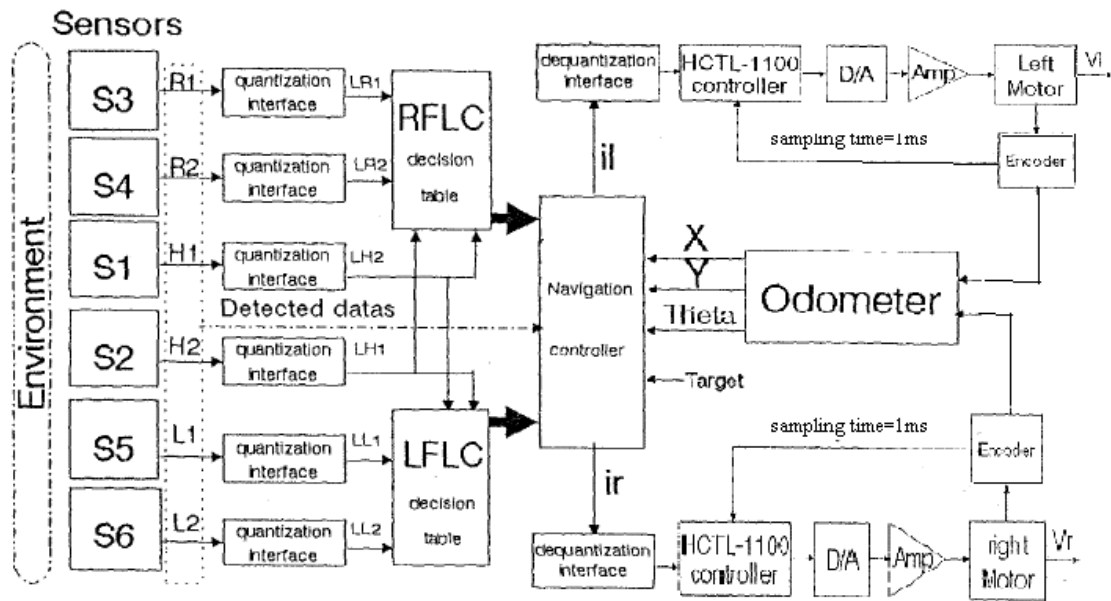
IF LH1 is BIG, LH2 is BIG, LR1 is BIG, LR2 is BIG
 THEN IL is BIG AND IR is SMALL.
 IF LH1 is BIG, LH2 is BIG, LR1 is BIG, LR2 is
 MIDDLE
 THEN IL is BIG AND IR is SMALL.
 IF LH1 is BIG, LH2 is BIG, LR1 is BIG, LR2 is
 SMALL
 THEN IL is BIG AND IR is SMALL.

...

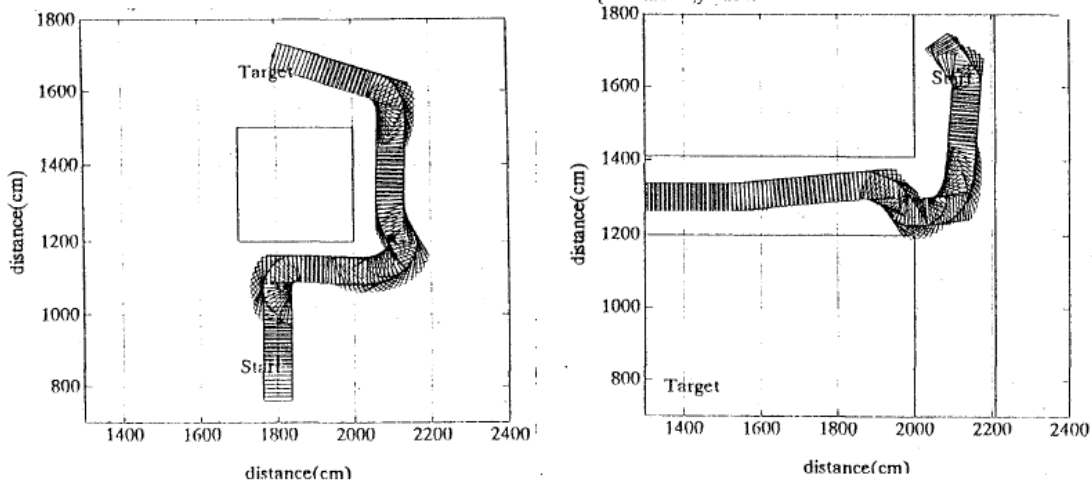
inputs				outputs	
LH1	LH2	LR1	LR2	IL	IR
4	4	4	0	4	0
4	4	4	1	4	0
4	4	4	2	4	0
...

شکل 10-11 جدول تصمیم‌گیری

در شکل 10-12 ساختار يك سیستم ناوبری فازی نشان داده شده است. در این شکل در ابتدا داده ها توسط حسگرها جمع آوری می شود و سپس کوانتیزه می شود. جداول تصمیم گیری نیز پیشتر به صورت off-line محاسبه می شود. کنترلر ناوبری داده های حسگرها، موقعیت مقصد، مکان روبات متحرك بررسی کرده و بر اساس آن تصمیم می گیرد که چه روشی بایستی استفاده گردد در شکل 10-13 نمونه ای از مسیری که روبات برای رسیدن به هدف طی می کند را نشان می دهد.



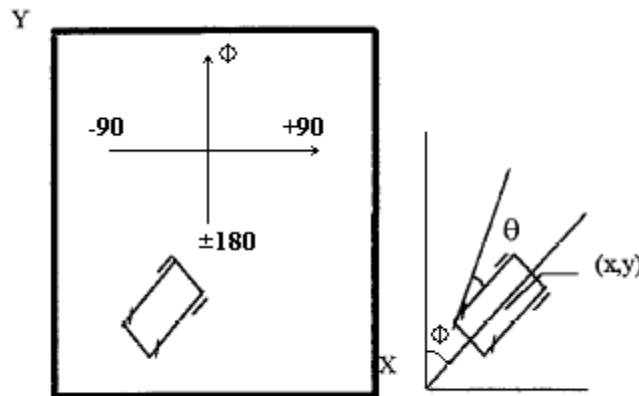
شکل 10-12 ساختار يك سیستم ناوبری فازی



شکل 10-13 مسیری که روبات برای رسیدن به هدف طی می کند

11. حل مسئله پارک دنده عقب و روش مدل سازی آنرا به طریقه K. Tanaka تشریح کنید.

صورت مسئله عبارتست از از عقب راندن کامیون برای بارگیری در محل معین. هدف طراحی سیستمی است که هدایت کامیون را از يك موقعیت اولیه بر عهده گرفته و کامیون را به محل بارگیری هدایت کند. ورودی‌هایی که سیستم می‌گیرد، عبارتند از: (x_t, y_t) , موقعیت فعلی کامیون و (ϕ) زاویه‌ای که با محور عمودی می‌سازد. خروجی سیستم نیز θ است که مقدار چرخش فرمان کامیون را برای نزدیک شدن به هدف تعیین می‌کند. مقادیر ورودی و خروجی هر دو فازی می‌باشند. مشخصات دنیایی که مسئله در آن حل می‌شود و جهت‌ها در شکل ۱-۱۱ نشان داده شده است:



شکل ۱-۱۱: مشخصات فضای مسئله

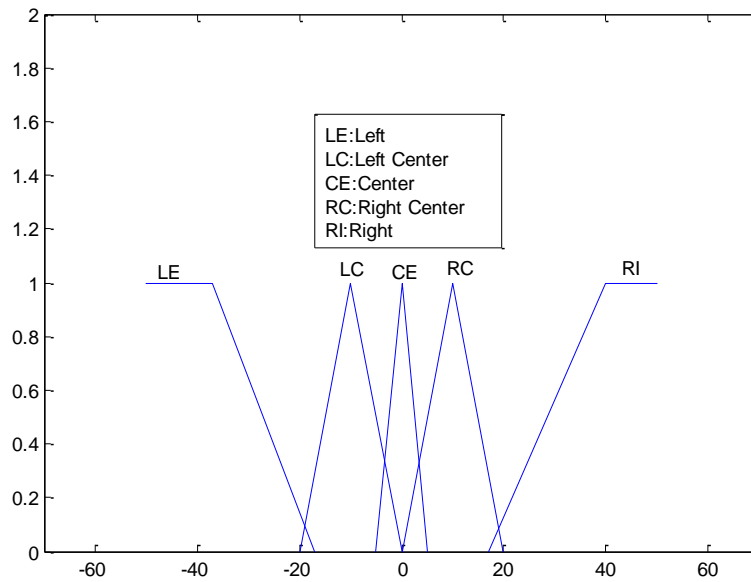
در حل مسئله توجه به نکات زیر ضروری است:

- y_t , یعنی فاصله عمودی از محل بارگیری, برای تعیین سرعت کامیون به کار می‌رود و به کنترل فرمان ربطی ندارد. اگر y_t از حدی کوچکتر باشد, باید سرعت متناظر با آن, کم شود, ولی, اگر سرعت از يك حد مورد نظر بزرگتر شد, سرعت کامیون از سرعت فعلی بالاتر نخواهد رفت. در این صورت, بعد از تنظیم شدن مختصات x فقط کافیسست که يك دنده عقب مستقیم برویم تا فاصله عمودی نیز جبران شود. حدود کردن سرعت کامیون به این دلیل است که امکان برخورد کامیون به اشیای اطراف وجود دارد.

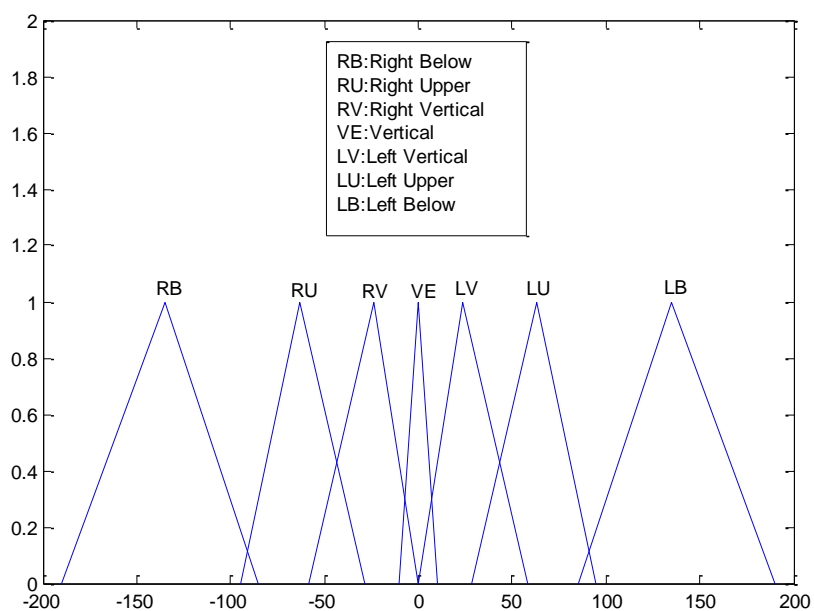
- میزان چرخش فرمان کامیون در بازه $[-30^\circ, +30^\circ]$ خواهد بود.

- میزان تغییرات x در بازه $[-50, +50]$ و میزان تغییرات ϕ در بازه $[-180, +180]$ می باشد.

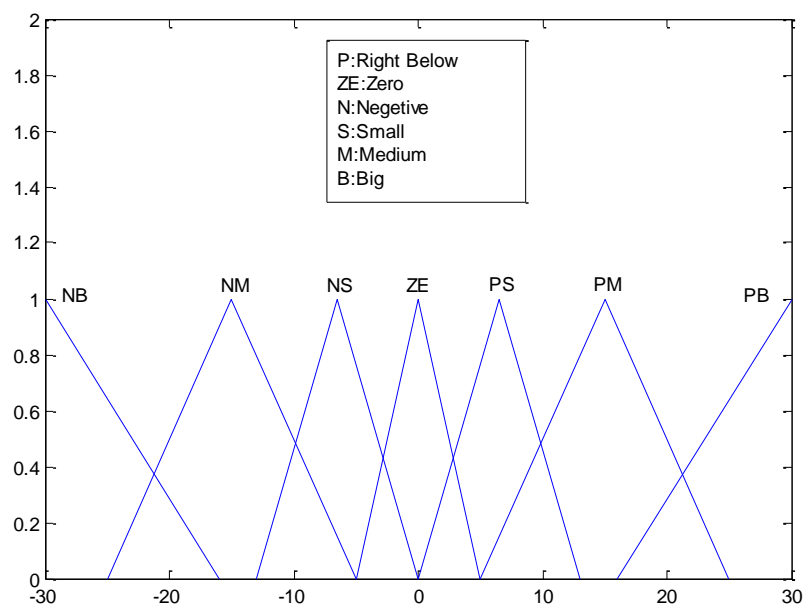
اولین قدم در حل این مسأله، استخراج توابع عضویت متغیرهای ورودی و خروجی است. در شکل ۲-۱۱ و ۳-۱۱ و ۴-۱۱ این توابع عضویت نمایش داده شده اند.



شکل ۲-۱۱: توابع عضویت متغیر x_t



شکل ۱۱-۳: توابع عضویت متغیر ϕ



شکل ۱۱-۴: توابع عضویت متغیر θ

حال نوبت به طراحی Rule Base می‌رسد. حالت شروع مسأله $(x_{t0}, y_{t0}, \emptyset_{t0})$ است و حال نهایی $(0, 0, 90)$ است. نمونه‌ای از قوانین کنترل‌کننده مورد نظر به صورت زیر است:

- اگر کامیون سمت راست و با فاصله زیاد از محل بارگیری باشد و جلوی کامیون به سمت پایین و راست باشد، فرمان باید کمی به سمت چپ چرخانده شود، یعنی:

IF x_t is LE and \emptyset is RB THEN θ is PS

۳۵ قانون استنتاجی برای این سیستم در جدول ۱-۱۱ آمده است.

جدول ۱-۱۱: قوانین استنتاج (جدول قواعد)

	LE	LC	CE	RC	RI
RB	PS	PM	PM	PB	PB
RU	NS	PS	PM	PB	PB
RV	NM	NS	PS	PM	PB
VE	NM	NM	ZE	PM	PM
LV	NB	NP	NS	PS	PM
LU	NB	NB	NM	NS	PS
LB	NB	NB	NM	NM	NS

برای مثال قوانین (RU, RC) و (RV, CE) را در نظر بگیرید:

(RU, RC) : IF x_t is RC AND \emptyset is RU THEN θ is PB

یعنی اگر کامیون در سمت راست و نزدیک بارگیری بوده و قسمت جلوی آن بطرف راست و بالا است، فرمان باید به مقدار زیادی به سمت چپ بچرخد.

(RV, CE) : IF x_t is CE AND \emptyset is RV THEN θ is PB

اگر کامیون تقریباً روبروی محل بارگیری بوده و قسمت جلوی آن به سمت بالا و کمی راست باشد، فرمان باید کمی به سمت چپ بچرخد.

12. روش طراحی کنترلر فازی Tanaka – Sugeno را تشریح نمایید.

این روش طراحی کنترلر شرایط پایداری را برقرار میکند و از سه مرحله تشکیل شده است:

مرحله اول: در این مرحله به وسیله اتصال مدل فازی محیط و کنترلر فازی آن یک سیستم فازی کامل به دست می آوریم.

به عنوان مثال فرض کنید مدل فازی زیر داده شده است:

$$L^1: \text{if } x(k) \text{ is } A^1 \text{ and } u(k) \text{ is } B^1 \text{ THEN } x^1(k+1) = a^1x(k) + b^1u(k)$$

$$L^2: \text{if } x(k) \text{ is } A^2 \text{ and } u(k) \text{ is } B^2 \text{ THEN } x^2(k+1) = a^2x(k) + b^2u(k)$$

برای این مدل فازی به راحتی میتوان کنترلر فازی زیر را به دست آورد:

$$L^1: \text{if } x(k) \text{ is } A^1 \text{ and } u(k) \text{ is } B^1 \text{ THEN } u^1(k) = f^1x(k)$$

$$L^2: \text{if } x(k) \text{ is } A^2 \text{ and } u(k) \text{ is } B^2 \text{ THEN } u^2(k) = f^2x(k)$$

در اینجا زیرسیستمهای R1 و R2 برای زیرسیستمهای L1 و L2 به همراه بازخوردهای حالت طراحی شده اند. به وسیله اتصال نوع B سیستم فازی کامل زیر به دست می آید:

$$s^{11}: \text{if } x(k) \text{ is } (A^1 \text{ and } A^1) \text{ and } u(k) \text{ is } (B^1 \text{ and } B^1) \text{ THEN } x^{11}(k+1) = (a^1 - b^1f^1)x(k)$$

$$2s^{11*}: \text{if } x(k) \text{ is } (A^1 \text{ and } A^2) \text{ and } u(k) \text{ is } (B^1 \text{ and } B^2) \text{ THEN } x^{12}(k+1) = (a^1 + a^2 - b^1f^2 - b^2f^1)x(k)/2$$

$$s^{11}: \text{if } x(k) \text{ is } (A^2 \text{ and } A^2) \text{ and } u(k) \text{ is } (B^2 \text{ and } B^2) \text{ THEN } x^{22}(k+1) = (a^2 - b^2f^2)x(k)$$

مرحله دوم: در این مرحله پارامترهای f_i مربوط به کنترلر را تعیین میکنیم. تعیین این

پارامتر پایداری زیرسیستمهای خطی سیستم کلی را تعیین میکند. این پارامترها را میتوان به راحتی میتوان به وسیله تئوری سیستمهای خطی پیدا کرد.

مرحله سوم: در نهایت ما پایداری سیستم کلی را بررسی میکنیم و اگر سیستم پایدار نبود،

به مرحله 2 بر میگردیم. میتوان مثالی را عنوان کرد که نشان دهد یک سیستم فازی، حتی اگر تمام زیرسیستمهای آن پایدار باشند ممکن است فازی نباشد.

مراجع:

1. Sugeno, M., and Takagi, T. (1983) "A New Approach to Design of Fuzzy Controller, *"Advances in Fuzzy Sets, Possibility Theory and Applications"*, pp. 325-334.
2. K.W. Wong, L.T. Kóczy, T.D. Gedeon, A. Chong, D. Tikk, *"Improvement of the Cluster Searching Algorithm in Sugeno and Yasukawa's Qualitative Modeling Approach"*.
3. Joongseon Joh, Reza Langari, Eun Tae Jeong and Won Jee Chung, *"A new Design Method for Continues Takagi-Sugeno Fuzzy Controller"*, IEEE 97.
4. Edwin Lughofer *"Online Adaptation of Takagi-Sugeno Fuzzy Inference Systems"*, Technical Report in Fuzzy Logic Laboratorium Linz-Hagenberg.

13. اندازه های فازی، مقایسه فازی و انتگرال های فازی را تشریح کنید.

ساگنو (1974) نظریه اندازه ها و انتگرال های فازی را معرفی کرد. یک اندازه فازی g روی یک مجموعه X (مجموعه مرجع با زیر مجموعه های E, F, \dots) شرایط زیر را ارضا می کند (X محدود است):

- 1) $g(\emptyset) = 0, \quad g(X) = 1$
- 2) If $E \subset F$, then $g(E) \leq g(F)$

یک اندازه فازی، یک اندازه ساگنو (یا یک -اندازه فازی) است اگر شرایط اضافی زیر را برای $\lambda > -1$ ارضا کند:

$$g_{\lambda}(E \cup F) = g_{\lambda}(E) + g_{\lambda}(F) + \lambda \cdot g_{\lambda}(E) \cdot g_{\lambda}(F)$$

مقدار λ می تواند با توجه به شرط $g(X) = 1$ محاسبه گردد:

$$\lambda + 1 = \prod_{i=1}^n (1 + \lambda g^i).$$

مثال محاسبه اندازه ساگنو:

مجموعه $X = \{a, b, c\}$ را در نظر بگیرید. مقادیر چگالی فازی اینگونه اند:

$$g^i = \begin{cases} 0.3 & \text{if } i = a, \\ 0.4 & \text{if } i = b, \\ 0.1 & \text{if } i = c. \end{cases}$$

مقدار λ را می توان با حل معادله زیر بدست آورد:

$$1 + \lambda = (1 + 0.3\lambda)(1 + 0.4\lambda)(1 + 0.1\lambda)$$

جوابها $\lambda = \{-16.8, 1\}$ هستند. با در نظر گرفتن شرط $\lambda > -1$ ، جواب $\lambda = 1$ قابل قبول می باشد. اندازه ساگنو می تواند اینگونه ساخته شود:

E	اندازه ساگنو
{a}	$g(\{a\}) = 0.3$
{b}	$g(\{b\}) = 0.4$
{c}	$g(\{c\}) = 0.1$
{a, b}	$g(\{a, b\}) = g(\{a\}) + g(\{b\}) + \lambda g(\{a\}) g(\{b\}) = 0.82$
{a, c}	$g(\{a, c\}) = g(\{a\}) + g(\{c\}) + \lambda g(\{a\}) g(\{c\}) = 0.43$
{b, c}	$g(\{b, c\}) = g(\{b\}) + g(\{c\}) + \lambda g(\{b\}) g(\{c\}) = 0.54$
{a, b, c}	$g(\{a, b, c\}) = g(X) = 1$

انتگرال فازی

انتگرال فازی (در مقالات همچنین: انتگرال ساگنو) می تواند بعنوان یک عملگر تجمع (aggregation) نگریسته شود. فرض کنید X مجموعه ای از عناصر باشد (مثلا ویژگی ها، حسگرها، طبقه بندی کننده ها). فرض کنید $h: X \rightarrow [0, 1]$. $h(x)$ درجه اعتقاد عنصر x را نشان می دهد (مثلا عضویت یک داده در یک دسته که توسط یک طبقه بندی کننده خاص تعیین شده است). انتگرال فازی h روی E (یک زیر مجموعه از X) با توجه به اندازه فازی g می تواند اینگونه محاسبه گردد:

$$\int_E h(x) \circ g = \sup_{\alpha \in [0,1]} [\alpha \wedge g(E \cap H_\alpha)];$$

With

$$H_\alpha = \{x | h(x) \geq \alpha\}.$$

مثال محاسبه انتگرال ساگنو

با فرض اینکه h بصورت زیر تعریف شده است:

$$h(x) = \begin{cases} 0.9 & \text{falls } x = a, \\ 0.6 & \text{falls } x = b, \\ 0.3 & \text{falls } x = c. \end{cases}$$

انتگرال ساگنو با توجه به اندازه ساگنو که در بالا تعریف گردید می تواند بدین صورت محاسبه گردد:

$$\begin{aligned} \int h \circ g_A &= \vee [(h(a) \wedge g_A(\{a\})), (h(b) \wedge g_A(\{a, b\})), (h(c) \wedge g_A(\{a, b, c\}))] \\ &= \max[\min(0.9, 0.3), \min(0.6, 0.82), \min(0.3, 1)] \\ &= \max[0.3, 0.6, 0.3] = 0.6 \end{aligned}$$

مقایسه فازی

در برنامه نویسی، مقایسه دو عدد به طرق گوناگون رایج است. مثلاً ممکن است بخواهیم بدانیم که عدد ورودی X مساوی مقدار خاصی است یا نه: تقریباً در همه زبانها می توانیم دستورالعملی به این صورت بنویسیم:

if $x = 200$ then (do something)

بطور مشابه در FLOPS می توانیم قانونی بنویسیم که همان کار را انجام دهد:

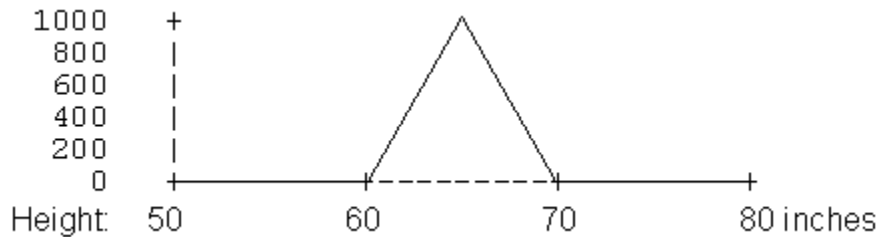
rule IF $x = 200$ THEN (do something)

در هر دو حالت اگر x حتی به میزان بسیار کمی تفاوت داشته باشد (مثلاً $x = 199.99$)، دستورالعمل اجرا نخواهد شد و قانون فعال (fire) نمی شود. در استدلال فازی، اغلب با معادله دقیق سروکار نداریم، بلکه خواهان این هستیم که بگوییم:

IF x is approximately 200 THEN (do something)

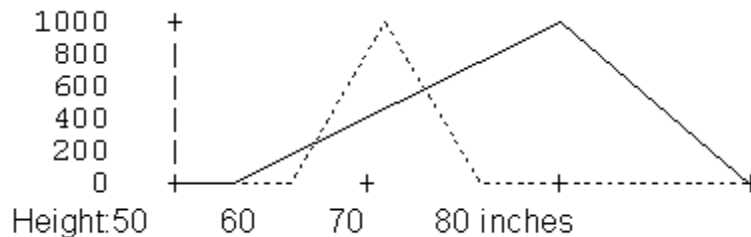
استفاده از اعداد فازی به ما اجازه این کار را می دهد. FLOPS مجموعه کاملی از مقایسه های عددی تقریبی دارد: $\sim <$ تقریباً کوچکتر، $\sim \leq$ تقریباً کوچکتر یا مساوی، $\sim =$ تقریباً مساوی و غیره. تنها شرط این است که حداقل یکی از اعداد مورد مقایسه، یک عدد فازی باشد.

عدد فازی شکل زیر، 65 فازی است. فرض کنید می خواهیم آزمایش کنیم که یک ورودی که مقدار آن 62.5 است، تقریباً مساوی 65 فازی می باشد یا نه. اگر به شکل زیر بنگریم، می توانیم ببینیم که درجه اعتقاد اینکه 62.5 به 65 فازی تعلق داشته باشد، حدود 500 است.



مقایسه تقریبی: "63 تقریباً 65 فازی است" با درجه اعتقاد 500 صحیح است.

فرض کنید دو عدد فازی را مطابق شکل زیر مقایسه می کنیم:



می بینیم که دو منحنی x و o در دو نقطه یکدیگر را قطع می کنند. درجه اعتقاد اینکه مقایسه برقرار است، مقدار بزرگتر اعتقاد در این دو نقطه تلاقی است یعنی 600.

14. استفاده از مفاهیم فازی در پایگاه های داده را مرور کنید.

هر چقدر که کاربرد تکنولوژی پایگاه داده از قلمرو ریاضیات قطعی (crisp) به قلمرو جهان واقعی منتقل می شود، نیاز به کنترل اطلاعات نادقیق مهمتر می شود چرا که یک پایگاه داده که بتواند اطلاعات نادقیق را کنترل کند باید نه تنها داده های خام را ذخیره کند بلکه باید اطلاعات مرتبطی را که امکان تفسیر داده ها در حوزه های بزرگتر و عمیقتری را بدهد نیز باید ذخیره سازد. مثلاً پرس و جوی (query) "کدام دانشجو جوان است و نمره های نسبتاً خوبی دارد" منظور واقعی کاربر را اینگونه بیان می سازد:

`SELECT * FROM STUDENT WHERE AGE < 19 AND GPA > 3.5`

چنین تکنولوژی، کاربردهای گسترده ای در حوزه هایی مانند معاینه، استخدام، سرمایه گذاری و غیره دارد. چون در این حوزه ها، اطلاعات ذهنی و غیر قطعی نه تنها رایج است بلکه بسیار مهم می باشد.

یکی از دغدغه های اصلی در طراحی و پیاده سازی پایگاه داده های فازی، کارایی است یعنی این سیستم ها باید به اندازه کافی سریع باشند تا بتوانند ارتباط و تعامل با کاربر را امکانپذیر سازند. بطور کلی، دو روش عملی برای اعمال فازی بودن در پایگاه های داده داریم:

1. اعمال پرس و جوهای فازی به پایگاه های داده کلاسیک.

2. افزودن اطلاعات فازی به سیستم.

داده اطلاعات می تواند بصورت زیر طبقه بندی شود:

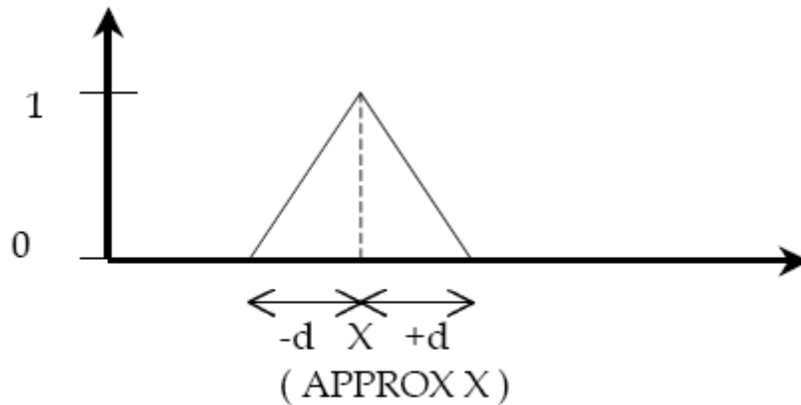
1. قطعی: هیچ ابهامی در اطلاعات نیست. مثلاً $X = 13$ یا دما $= 90$ درجه.

2. فازی: ابهامی در اطلاعات وجود دارد که خود می تواند به دو نوع تقسیم شود:

I. مقدار تقریبی: داده اطلاعاتی کلاً مبهم نیست و یک مقدار تقریبی است که معلوم

است و داده ها حول و حوش این مقدار قرار دارند. مثلاً $10 < X < 15$ که فرض

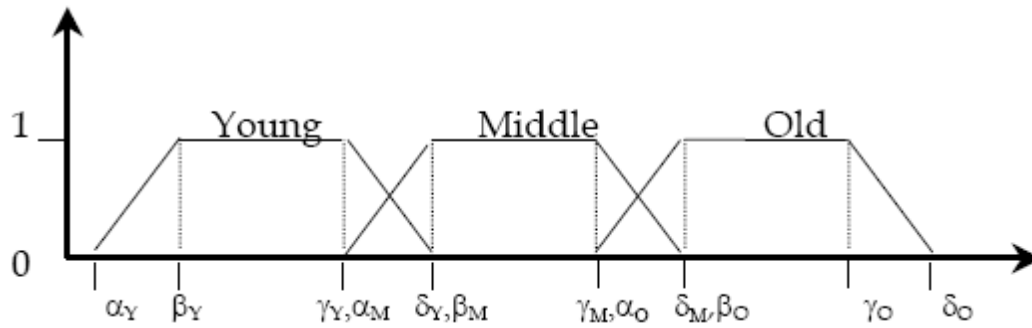
می شود یک توزیع امکان مثلثی شکل دارند. پارامتر d ، محدوده ای را نشان می دهد که اطلاعات در آن قرار می گیرند.



➤ متغیر زبانی: یک متغیر زبانی، متغیری است که جدا از نشان دادن یک عدد فازی، مفاهیم زبانی تفسیر شده در یک حوزه خاص را هم بیان می دارد. هر متغیر زبانی بصورت متغیری تعریف می شود که یا یک تفسیر فیزیکی دارد (سرعت، وزن ...) یا مقادیر عددی دیگری را اختیار می کند (حقوق، غیبت ها، نمرات ...).

آسانترین راه افزودن فازی بودن به مدل پایگاه داده، استفاده از پایگاه های داده رابطه ای کلاسیک و ایجاد یک واسط در آن است که بتواند پرس و جوی فازی را در پایگاه داده انجام دهد. یک محدودیت موجود در سیستم این است که چون ما مدل پایگاه داده را گسترش نمی دهیم و یک مدل جدید هم تعریف نمی کنیم، مدل زیرین پایگاه داده قطعی است و لذا فازی بودن می تواند تنها در پرس و جو خود را نشان دهد.

برای اعمال فازی بودن ما مجموعه/متغیر زبانی ها فازی را به حوزه های ویژگی/متغیرهای زبانی اضافه می کنیم. مثلاً در حوزه ویژگی سن، می توانیم مجموعه های فازی جوان، میانسال و پیر را تعریف کنیم. اینها بصورت زیر تعریف شده اند:



برای این کار، ما از یک پایگاه داده دانشجو استفاده می کنیم که یک جدول STUDENTS با ویژگی های زیر دارد:

a. Name b. Age c. Course b. Percentage c. Absences

Name	Age	Course	Percentage	Absences
Ankit	19	12	83	13
Anuj	17	10	80	9
Sumit	18	11	83	6
Rahul	19	12	56	12
Bishop	19	12	65	32
Neha	18	11	77	23
Malini	17	10	69	10
Rocky	16	9	79	13
Sandeep	19	12	75	6
Nagesh	19	12	83	6

در سطح دانش میانی، تنها به افزودن یک جدول نیاز داریم: LABELS با ساختار زیر:

LABELS

Label	Column_Name	Alpha	Beta	Gamma	Delta
-------	-------------	-------	------	-------	-------

این جدول برای ذخیره اطلاعات همه مجموعه های فازی تعریف شده روی حوزه های ویژگی بکار می رود. شرح هر ستون بدین صورت است:

➤ Label: کلید اولیه این جدول است و مقدار زبانی متناظر با مجموعه فازی را

ذخیره می کند.

- Column_Name: متغیر زبانی متناظر با مقدار زبانی را ذخیره می کند.
- Alpha, Beta, Gamma, Delta: محدوده مجموعه های فازی را ذخیره می کند.

موضوع اصلی در پیاده سازی این سیستم، تجزیه (Parsing) پرس و جوی فازی ورودی است. چون پایگاه داده زیرین، قطعی است یعنی هیچ داده فازی ذخیره نشده است، پرس و جوی INSERT تغییر نمی کند و نیازی به تجزیه ندارد. در طول تجزیه، پرس و جو تجزیه شده و به موارد زیر تقسیم می گردد:

- نوع پرس و جو: آیا پرس و جو، SELECT, DELETE یا UPDATE است.
- ویژگی های نتیجه: ویژگی هایی که فقط باید در پرس و جوی SELECT نشان داده شوند.
- جداول ورودی: جدول هایی که پرس و جو روی آنها اعمال می گردد.
- شرایط: شرایطی که باید قبل از انجام عملیات مشخص گردند.

بر اساس طبقه بندی داده های اطلاعاتی، ویژگی های درون پایگاه داده به دو نوع تقسیم می گردند.

- نوع 1: ویژگی فقط مقادیر قطعی را نگهداری می کنند.
- نوع 2: ویژگی، فازی است و می تواند یک مقدار قطعی، یک مقدار تقریبی یا یک مقدار زبانی بگیرد.

Name	Age	Course	Percentage	Absences
Ankit	OLD	12	GOOD	13
Anuj	MIDDLE	10	80	9
Sumit	18	11	83	LOW
Rahul	OLD	12	BAD	12
Bishop	19	12	65	HIGH
Neha	18	11	AVERAGE	23
Malini	17	10	69	10
Rocky	MIDDLE	9	79	13
Sandeep	APPROX 19	12	APPROX 75	APPROX 6
Nagesh	APPROX 19	12	83	APPROX 6

نوع عملیات و نحو (Syntax) آنها می تواند بصورت زیر باشد:

INSERT

INSERT INTO <TABLE> VALUES(<expr1>, ...)

DELTE

DELTE FROM <TABLE> [WHERE <COND1> [<CON><COND2>...]]

UPDATE

UPDATE <TABLE> SET VALUES <ATR1> = <expr1>
[,<ATR2> = <expr2>...]
[WHERE <COND1> [<CON><COND2>...]]