**طراحي كنترل كننده هاي فازي**

**مقدمه**

در سال ١٩٧٥ ممداني و اصيليان براي اولين بار يك کوره سيمان را با منطق فازي كنترل كردند. در سال ١٩٧٨ هولمبلاد و اوسترگارد اولين كنترل كننده فازي را براي كنترل يك فرآيند صنعتي كامل، يعني كوره سيمان به كار بردند. **از** آن پس بود كه كنترل كننده هاي فازي در بسياري از دستگاهها از قبيل ماشين لباس شويي، دوربين فيلمبرداري، پلوپز و … و فرآيندهاي صنعتي مانند مترو و روباتيك و … به كار برده شد. كنترل فازي يك روش كنترل بر اساس منطق فازي است. در حقيقت اگر منطق فازي را به طور ساده " محاسبه با كلمات به جاي اعداد " بناميم، كنترل فازي را مي توان " كنترل با جملات به جاي معادلات " ناميد. يك كنترل فازي شامل يك سري قواعد است. براي مثال براي يك كنترل كننده فازي نوعي داريم:

١ – اگر خطا منفي بود و مشتق خطا منفي بود، آنگاه خروجي خيلي منفي مي باشد.

٢ – اگر خطا منفي بود و مشتق خطا صفر بود، آنگاه خروجي كمي منفي باشد. و...

به مجموعه قواعد يك كنترل كننده فازي، "پايگاه قواعد" مي گوييم. قواعد به شكل آشناي " اگر – آنگاه" مي باشند.

با مجموعه قواعدي نظير فوق، كامپيوتر قادر خواهد بود با در نظر گرفتن خطا و مشتق خطا برنامه مربوطه را اجرا كرده و سيگنال **كنترلي** (خروجي كنترل كننده) را محاسبه كند. اين نوع كنترل را مي توان اجرای عمل كنترل توسط يك اپراتور دانست. يك اپراتور با زير نظر گرفتن رفتار سيستم (ورودي به اپراتور) و طبق تجربيات و دانسته هاي خود (پايگاه قواعد] يك فرمان كنترلي (خروجي) مي دهد يا پارامترهاي يك كنترل كننده را تنظيم مي نمايد. براي طراحي فازي روش هايی مانند مكان ريشه، طراحي بر اساس پاسخ فركانسي، جابجايي قطبها و غيره وجود ندارد. كنترل كننده هاي فازي در شكلهاي مختلف كنترل ظاهر شد ه اند كه يكي از پرکاربردترين آنها كنترل

مستقيم است که درشكل زیر نشان داده شده است.



**شكل ‏2‑2:**  نمايش بلوکی استفاده از کنترل کننده فازی به صورت مستقيم

در اين حالت كنترل كننده فازي در مسير پيشروی يك سيستم كنترلي فيدبك دار قرار مي گيرد. خروجي فرآيند با ورودي مرجع مقايسه **شده** و در صورت وجود خطا كنترل كننده يك سيگنال كنترلي مطابق استراتژي كنترلي خود توليد مي كند. لازم به ذكر است كه ورودي كنترل كننده مي تواند مشخصه هاي ديگري از سيستم نظير مشتق خطا، انتگرال خطا و يا ترکيبی از آنها نيز باشد.

**ساختار يك كنترل كننده فازي**

يك كنترل كننده فازي از چهار بخش اصلي تشكيل **شده** است. فازي كننده، پايگاه قواعد، بخش تصميم گيري، غير فازي كننده. شكل زیر نمايي از يك كنترل كننده فازي را نمايش مي دهد. معمولاً قبل و بعد از كنترل كننده فازي همانند شكل زیر به ترتيب پيش پردازنده و پس پردازنده استفاده مي شود. در بخشهاي بعد، اين قسمتها را توضيح خواهيم داد.

**** **شكل ‏2‑3:**  دياگرام بلوکی ساختار کنترل کننده فازی

**پيش پردازش**[[1]](#footnote-2)

ورودي يك كنترل كننده فازي مقادير قطعي است كه توسط وسايل اندازه گيري و سنسورها به دست آمده است. معمولاً اين مقادير نياز به تغييراتي **قبل** از ورود به كنترل كننده فازي دارند. اين تغييرات توسط بخش پيش پردازنده فراهم آورده مي شود.

مثالهايي از پيش پردازنده ها عبارتند از:

-كوانتيزه كردن همراه با نمونه برداري يا گرد كردن به مقادير صحيح

-نرماليزه كردن يا مقياس بندي به يك بازه خاص يا استاندارد

-فيلتر كردن به منظور حذف نويز

-مشتق گيري و انتگرال گيري

وقتي ورودي كنترل كننده فقط خطا باشد استراتژي كنترلي يك كنترل ايستا (static) مي باشد. اما يك كنترل كننده **ديناميك**(dynamic) نيازمند ورود يهاي ديگر نظير مشتق، انتگرال يا مقادير قبلي اندازه گيري شده مي باشد.

اين مقادير ابعاد كنترل كننده را زيادتر كرده و در نتيجه تعداد قواعد بيشتري نياز است كه طراحي آن را مشكل مي كند .

**فازي كننده**[[2]](#footnote-3)

اولين بلوك داخل كنترل كننده، فازي كننده مي باشد، كه در آن درجه تعلق مقادير ورودي به تابع تعلقهای مختلف محاسبه مي شود. براي اين كار چندين روش موجود است. روش اول اين است كه متغير را به صورت تك مقداري ١، به صورت فازي در آوريم. در اين حالت تابع تعلق عضويت متغير در يك نقطه " 1" و در بقيه نقاط " 0" مي باشد. شكلa-۳زير نمونه اي از اين گونه را نشان مي دهد.



**شكل ‏2‑4:** (a) تابع تعلق تک مقداری(b) تابع تعلق مثلثی با محدوده ٢

در روش ديگري مي توان آن را به صورت يك منحني نمايش د اد. مثلاً فرض كنيد انحراف معيار در حالت غير فازي به علت وجود نويز يا خطاي اندازه **گيري** برابر باشد. در آن صورت تابع فازي متغير به صورت مثلثي متساوي الساقين كه ميانه قاعده آن در مقدار اندازه گرفته شده بود، طول قاعده آن برابر2 بوده و بلنديي برابر ١ دارد، تعريف مي شود.

**پايگاه قواعد** [[3]](#footnote-4)

**پايگاه قواعد به مجموعه " اگر – آنگاه " فازي گفته مي شود كه قسمت هوشمند فازي را تشكيل مي دهد. براي تنظيم قواعد چهار روش وجود دارد:**

**-دانش مهندسي كنترل و دانش خبره**

**-رفتار اپراتور در موقع كنترل**

**-بر پايه مدل فازي فرآيند (شناسايي فازي سيستم)**

**-بر پايه آموزش خود سازمانده**

**آنچه در اين پروژه انجام شده است روش آخر مي باشد كه پايگاه قواعد با استفاده از الگوريتم ژنتيكي آموزش مي بيند و بهينه مي شود.**

**قواعد ممكن است داراي چند متغير هم در قسمت شرط و هم در قسمت نتيجه شان باشند. در اين حالت كنترل ك ننده مي تواند به سيستمهاي تك ورودي – تك خروجي (SISO) و هم چند ورودي – چند خروجي(MIMO ) اعمال شود. يك كنترل كننده نوعي ممكن است از خطا، مشتق خطا و انتگرال خطا، با هم استفاده كند. اما به اين حالت چند ورودي نمي گوييم چون هر سه سيگنال ورودي از اندازه گيري خطا به دست آمده اند.**

**قالبهاي قواعد**[[4]](#footnote-5)

هر كنترل كننده فازي از تعدادي " اگر آنگاه " به عنوان پايگاه قواعد استفاده مي كند. اين قواعد به شكلهاي مختلف داده مي شود. فرض كنيد يك كنترل كننده فازي داراي دو ورودي خطا و يك سيگنال به عنوان خروجي توليد مي كند. آنگاه قالب نمايش end-user به شكل زير مي باشد:

1 – if error is Neg error dot is Neg then OutPut is NB

:

:

9 -. ... .

نامهاي Pos، Zero، Neg مربوط به مجموعه هاي فازي براي هر دو ورودي مي باشند و NB، NS، PB، PS(به ترتيب منفي بزرگ، منفي کوچک ،مثبت بزرگ، مثبت کوچک) مجموعه هاي فازي خروجي مي باشند.

همان قواعد بالا در قالب فشرده تري به نام " قالب نسبي " نيز نمايش داده مي شود.

****

اين شكل براي كاربران با تجربه كه مي خواهند نگاهي سريع به مجموعه قواعد داشته باشند، مناسب مي باشد. در اين حالت پيش فرض اين مي باشد كه بين ورودي اتصال دهنده " and" و بين قواعد از " or" استفاده شده است.

از قالب فشرده تري با نام "قالب جدولي" نيز استفاده می شود که در زير آورده شده است:

****

**شكل ‏2‑5:**  قالب جدولی پايگاه قواعد

در اين حالت خالي بودن خانه جدول به معني عدم وجود قاعده در حالت مربوط به آن خانه است. همچنين بايد توجه داشت كه وقتي تعداد ورودي ها بيشتر از ٢ مي شود، ابعاد جدول هم بيشتر از ٢ خواهد شد و در نتيجه در نمايش آن دچار اشكال خواهيم شد.

وقتي كنترل كننده چند خروجي است مي توان قواعد آن را به چند قاعده يك خروجي تبديل كرد و در هر خانه جدول، چند خروجي را نشان داد. آخرين قالب نمايش، قالب گرافيكي مي باشد. كه در آن نمايش تابع تعلقها هم امكان پذير مي باشد. در اين حالت نشان دادن نحوه استنتاج به راحتي صورت مي پذيرد. اين قالب در شکل زیر نشان داده شده است.



**شكل ‏2‑6:** قالب گرافيکی نمايش پايگاه قواعد.

**مجموعه جهاني**[[5]](#footnote-6)

عناصر مجموعه فازي از يك مجموعه مرجع به نام مجموعه جهاني گرفته مي شوند. مجموعه جهاني شامل همه عناصري است كه مي توان براي سيستم متصور شد. قبل از طراحي تابع تعلق لازم است مجموعه جهاني براي ورودي و خروجي در نظر گرفته شوند. اين كار در حقيقت پيدا كردن ماكزيمم و مينيمم ورودي ها و خروجي هاست. تا همه مقادير ورودي اندازه گيري شده و خروجي اعمال به سيستم توسط كنترل كننده فازي قابل فهم باشند. نكته ديگر اين است كه توابع عضويت ورودي مي توانند گسسته يا پيوسته باشند. توابع عضويت پيوسته روي مجموعه جهاني پيوسته و توابع عضويت گسسته روي مجموعه جهاني گسسته عمل مي كنند. در حالت گسسته مجموعه جهاني يك بردار است كه وروديها بايد حتماً عضو آن باشند كه همان طور كه گفته شد پيش پردازش انجام اين كار را بر عهده مي گيرد.

**طراحي توابع عضويت**

هر عنصر مجموعه جهاني با يك درجه عضو توابع عضويت مي باشند. اشكال مختلف تابع عضويت در شكل زیر نشان داده شده است.



**شكل ‏2‑7**: مثالهايی از توابع عضويت: (a) تابع z، (b) گوسين، (c) تابع s ،(d-f) حالتهاي مختلف مثلثي، (g-i) حالتهاي مختلف ذوزنقه ای،(j) گوسين تخت، (k) مستطيلی، (l) تك مقداري

يكي از مسائلي كه در زمان طراحي بايد حل شود نحوه ساختن توابع عضويت است. در حقيقت با دو سئوال اساسي روبرو هستيم:

١) چگونه مي توان شكل مجموعه ها را به دست آورد؟

٢) چند مجموعه بايد تعريف شود؟

طبق قواعد سرانگشتي داريم:

* مجموعه ها بايد آنقدر پهن باشند كه نويز در اندازه گيري را شامل شوند.
* همپوشاني مجموعه ها بايد به اندازه كافي باشد. در غير اين صورت ممكن است كنترل كننده به حالتهايي برخورد كند كه خروجي خوبي نمي دهد.

براي پاسخ به سئوالات مطرح شده مي توان از تجربيات اپراتور استفاده كرد كه بعضي اوقات اين نيز مشكل است. در اين حالت پيشنهاد مي شود:

* با مجموعه اي مثلثي شروع كنيد: همه توابع عضويت براي ورود ي ها و خروجي ها بايد مثلثي متقارن با پهناي مساوي باشند.
* همپوشاني حداقل ٥٠ % باشد: در حالت اوليه پهناي هر مجموعه طوري انتخاب شود كه هر عضو مجموعه جهاني حداقل عضو دو تابع عضويت (با درجه غيرصفر ) باشد، بجز عناصر دو طرف محدوده مجموعه جهاني كه ممكن است فقط عضو يك مجموعه باشند. در حقيقت فاصله بين دو تابع عضويت موجب مي شود كه به ازاي عناصر آن فاصله هيچ قاعده اي فعال نشود.

**موتور استنتاج**[[6]](#footnote-7)

به دليل اينكه هر پايگاه قواعد فازي در عمل شامل بيش از يك قاعده مي شود، سئوال اساسي اين است كه چگونه مي توان از روي يك مجموعه از قواعد نتيجه گيري كرد. دو روش براي نتيجه گيري وجود دارد. استنتاج مبتني بر ترکيب قواعد و استنتاج مبتنی برقواعد جداگانه.

o استنتاج مبتني بر تركيب قواعد: در اين نوع استنتاج، تمامي قواعد موجود در پايگاه فازي در يك رابطه فازي در U × V تركيب شده و آنگاه بديده يك قاعده " اگر آنگاه" فازي تنها نگريسته مي شود.

اگر R(l) يك رابطه فازي در U × V يعني

و M قاعده داشته باشيم، از آنچه پيشتر ديديم داريم:

|  |  |
| --- | --- |
|  | **(‏2‑1)** |

كه× يك عملگر–t نرم ميباشد. اگر بين قواعد رابطه اجتماع را بپذيريم، آنگاه داريم:

|  |  |
| --- | --- |
|  | **(‏2‑2)** |
| μQ (x,y) μRu1 (x,y) ...μRuM (x,y) | **(‏2‑3)** |

كه + نشانه s- نرم است

اين تركيب ممداني ناميده مي شود.

اگر بين قواعد رابطه اشتراك را بپذيريم، تركيب گودل بدست مي آيد:

|  |  |
| --- | --- |
|  | **(‏2‑4)** |
| μQ(x,y) μRu1 (x,y)...μRuM (x,y) | **(‏2‑5)** |

كه × نشانه t- نرم مي باشد.

o استنتاج مبتني بر قواعد جداگانه: در اين حالت هر قاعده در پايگاه قواعد فازي يك خروجي فازي را معين كرده و خروجي نمايي تركيب Mخروجي جداگانه مجموعه اي فازي خواهد بود. عمل تركيب را با اجتماع يا اشتراك مي توان انجام داد. در اين حالت تابع تعلق خروجي را براي هر قاعده به طور جداگانه و از طريق مودس پوننس تعميم يافته به دست آورده، سپس بسته به استفاده از اجتماع يا اشتراك به ترتيب از تركيب ممداني يا گودل براي تركيب M قاعده استفاده مي كنيم.

با و جود دو نوع استنتاج گفته شده و استلزامهاي مختلف و همچنين عملگرهاي مختلف اين سئوال در طراحي مطرح مي شود كه كداميك از اين گزينه ها را انتخاب كنيم. در حالت كلي سه معيار زير را مي بايست در نظر گرفت:

* معناي شهودي: به عنوان مثال اگر يك انسان خبره معتقد باشد كه قواعد از هم مستقل هستند از اجتماع براي تركيب قواعد استفاده مي شود.
* راندمان محاسباتي: فرمولي در رابطه تابع عضويت خروجي به دست مي آيد كه پياده سازي محاسباتي آن ساده باشد.
* ويژگيهايي كه به صورت تجربي بدست مي آيند.

موتور استنتاج كه معمولاً در سيستم فازي استفاده مي شوند عبارتند از:

١) موتور استنتاج حاصل ضرب ٢) موتور استنتاج مينيمم

اگر از استنتاج مبتني بر قواعد جداگانه با اجتماع و استلزام ممداني و ضرب جبري براي t- نرمها

و max براي s - نرمها استفاده كنيم موتور حاصل ضرب به صورت زير به دست مي آيد:

|  |  |
| --- | --- |
|  | **(‏2‑6)** |

اگر استنتاج مبتني بر قواعد جداگانه با تركيب اجتماع، استلزام مينيمم ممداني، عملگر min براي- - t

نرمها و max براي s - نرمها استفاده كنيم موتور استنتاج مينيمم بدست مي آيد:

|  |  |
| --- | --- |
|  | **(‏2‑7)** |

**غير فازي ساز [[7]](#footnote-8)**

مجموعه فازي نتيجه استنتاج بايد تبديل به يك عدد غير فازي شود تا بتوان خروجي متناسب با آن را به عنوان سيگنال كنترل به فرآيند اعمال كرد. اين عمل را غيرفازي سازي مي نامند. در شكل ۷ مقدار روي محور x كه با رنگ سفيد علامت دار شده است، سيگنال كنترل بدست آمده از غيرفازي كردن نتيجه استنتاج مي باشد.دراين حالت خروجی کنترل کننده به ازای خطای ۵۰-، ۲۵ - به دست آمده است.



شکل ۷: استفاده از غيرفازی ساز مرکز جرم برای کنترل کننده تک ورودی تک خروجی

روشهاي مختلفي براي غيرفازي كردن وجود دارد:

 مركز جرم[[8]](#footnote-9): عددغيرفازي خروجي،u، تحت مركز جرم مجموعه فازي به دست آمده محاسبه

مي شود:

|  |  |
| --- | --- |
|  | **(‏2‑8)** |

كه xi نقاط درون مجموعه جهاني است.

در حقيقت در اين روش ميانگين وزني عناصر نتيجه استنتاج محاسبه مي شود. در حالت پيوسته

انتگرال به جاي جمع قرار مي گيرد. به اين روش مركز سطح نيز گفته مي شود.

مركز جرم براي حالت تك مقداري[[9]](#footnote-10): اگر تابع عضويت نتيجه تك مقداري باشد، مقدار خروجي برابر:

|  |  |
| --- | --- |
|  | **(‏2‑9)** |

مي باشد كه موقعيت" si تك مقدار i " ام در مجموعه جهاني مي باشد.

 دو بخشي سطح[[10]](#footnote-11): اين روش مقداري را برمي گرداند كه مقدار سطح زير منحني تابع عضويت را به دو قسمت مساوي تقسيم مي كند. در حالت پيوسته:

|  |  |
| --- | --- |
|  | **(‏2‑10)** |

min كوچكترين مقدار مجموعه جهاني و max بزرگترين مقدار آن مي باشد. اين روش محاسبات پيچيده اي دارد و همچنين در حالت گسسته تعريف نمي شود.

متوسط ماكزيمم[[11]](#footnote-12): در اين روش نقطه اي با بالاتر ين درجه عضويت در تابع تعلق خروجي استنتاج انتخاب مي شود. اگر چند نقطه با بالاترين درجه عضويت وجود داشت متوسط آنها را به عنوان سيگنال خروجي انتخاب مي كنيم. در اين روش محاسبات نسبتاً مناسب است.

**پس پردازنده**[[12]](#footnote-13)

در مواقعي كه نياز داريم خروجي در يك محدوده استاندارد قرار گيرد يا مي خواهيم تبديل مقياس انجام دهيم از پس پردازش روي سيگنال خروجي كنترل كننده فازي استفاده مي كنيم.

نتایج شبیه سازی مقاله:

مشاهده می شود که نتایج شبیه سازی با نتایج مقاله همخوای دارد.

1. Preprocessing [↑](#footnote-ref-2)
2. Fuzzifier [↑](#footnote-ref-3)
3. Rule base [↑](#footnote-ref-4)
4. Rule formats [↑](#footnote-ref-5)
5. Universe [↑](#footnote-ref-6)
6. Inference Engine [↑](#footnote-ref-7)
7. Defuzzifier [↑](#footnote-ref-8)
8. Centre Of Gravity [↑](#footnote-ref-9)
9. Centre Of Gravity method for Singletons [↑](#footnote-ref-10)
10. Bisector of Area [↑](#footnote-ref-11)
11. Mean Of Maxima [↑](#footnote-ref-12)
12. Postprocessing [↑](#footnote-ref-13)