

مدل سازی و پیاده سازی سخت افزاری کنترلر مسیر ربات توسط یک سیستم هیبرید

علی اکبر مصدق عنبران ، سعیده عیسی زاده

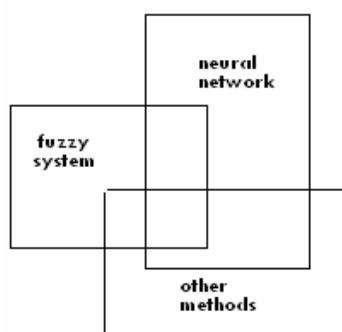
آزمایشگاه رباتیک موسسه آموزش عالی سجاد

E-mail: ali_mosaddegh2000@yahoo.com saide_issazade2006@yahoo.com

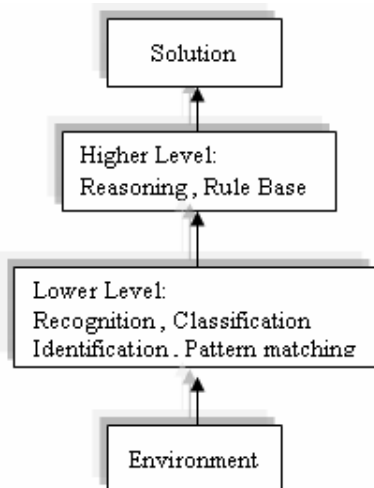
چکیده : هدف این مقاله معرفی یک سیستم هیبرید با استفاده از روش های حسابگری نرم همچون شبکه های عصبی و کنترل فازی است که از آن برای کنترل مسیر یک ربات استفاده می شود. بر اساس شبیه سازی و پیاده سازی های صورت گرفته بخش های مختلف این ربات مدل می شوند. در این سیستم می توان به عملکرد بالایی در مسائل مربوط به کنترلر و مکانیزم های مربوط به آن رسید که با به کارگیری سایر کنترلرها، به این حد مطلوب نمی توان دست یافت.

کلید واژه : ربات، سیستم هیبرید، شبکه عصبی، کنترل فازی.

۱- مقدمه

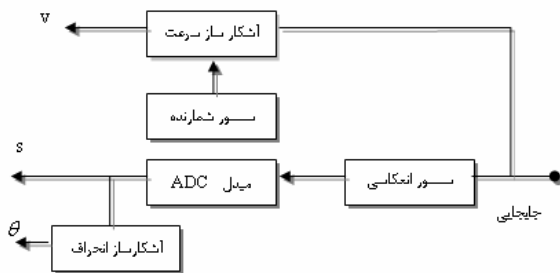


شکل ۱: همپوشانی و ترکیب چند روش مختلف



شکل ۲: مدل سلسله مراتبی دو طبقه رو به جلو

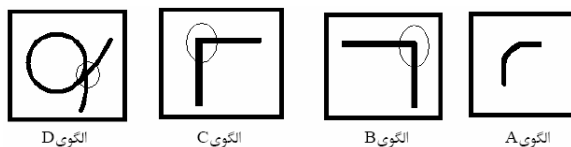
با متحول شدن دنیای علم و گسترش دانش بشری، بررسی و حل بسیاری از مسائل پیچیده، مستلزم پردازش حجم زیادی از اطلاعات و یا متغیرهای بسیار است. در این مقاله با ترکیب چند روش مختلف، یک سیستم هیبرید برای برخورد با این گونه مسائل پیشنهاد می شود (شکل ۱). در مدل پیشنهادی از یک روند سلسله مراتبی^۱ دو طبقه رو به جلو استفاده می شود. اطلاعات از طبقه اول بدون پسخور^۲ وارد طبقه بالایی می شود. طبقه اول دارای ۳ زیر ماژول شامل: پردازش ابتدایی داده ها، استخراج مشخصه ها و تشخیص و طبقه بندی الگوها است. طبقه دوم نیز دارای بخشهای مبتنی بر قانون^۳ می باشد (شکل ۲). در این طرح، عملکرد سیستم در ماژول هایی که به یکدیگر مرتبط هستند ساختار بندی و ارتباط ماژول ها با یکدیگر مشخص می شوند. ورودی و خروجی و روند حل مساله برای هر یک از ماژول ها مشخص می گردد و سیستم به صورتی که بتواند توسط جریان ورودی داده ها، به صورت داخلی و خود کنترل باشد، پیکر بندی می شود [۱].



شکل ۴: بخش های بلاک آشکار ساز

۲-۳- سیستم کنترل هیبرید

در آزمایش این سیستم، از مسیرهایی شامل تلفیق ۴ الگوی مشخص شکل (۵) استفاده شده است. مطابق شکل (۶)، در ابتدا ورودی s به ساختار شبکه عصبی MLP^۶ جهت طبقه بندی الگوها^۷ وارد می شود. این شبکه دارای ۱۲ نورون ورودی به ازاء هر یک از سنسورهای انعکاسی می باشد، که به عنوان بافر عمل می کنند. تعداد نورون های لایه میانی ۱۰۰ عدد می باشد. این تعداد با توجه به شبیه سازی ها و پاسخ خروجی شبکه و نیز سرعت مطلوب آن تعیین گردیده است. لایه آخر شامل ۲ نورون است که خروجی آنها به صورت باینری، برای تشخیص و تفکیک ۴ الگوی در نظر گرفته شده بکار می رود.

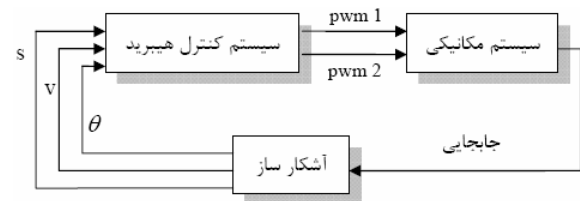


شکل ۵: الگوهای تشکیل دهنده مسیر حرکت ربات

در هر لحظه با اعمال ورودی جدید سنسورها (s)، شبکه عصبی یکی از ۴ الگوی آموزش داده شده را پس از اعمال به بلاک تطبیق الگو^۸ تشخیص می دهد و سپس بلاک تصمیم گیری^۹، کنترلر بهینه جهت کنترل ربات در الگوی تشخیص داده شده را انتخاب می کند. در الگوی A (شامل مسیر یکنواخت و انحنادار)، که بیش از ۹۰٪ طول مسیر را به خود اختصاص می دهد از یک کنترلر فازی جهت هدایت ربات استفاده شده است. مجموعه قوانین فازی این کنترلر هوشمند، با توجه به ۳ حالت تعریف شده برای سرعت (کم، متوسط، زیاد) و انحراف (به راست، متوسط، به چپ) دارای ۹ قانون می باشد. با اعمال ورودی های سرعت (v) و انحراف (θ) به بلاک فازی ساز^{۱۰} و استفاده از توابع عضویت بهینه شده توسط نرم افزار شبیه ساز، خروجی های فازی

۲- مدل سازی ربات

در یک مدل کلی مطابق شکل (۳)، ربات از دو بخش سیستم مکانیکی و سیستم کنترل هیبرید تشکیل می شود.



شکل ۳: مدل کلی ربات با استفاده از سیستم کنترل هیبرید

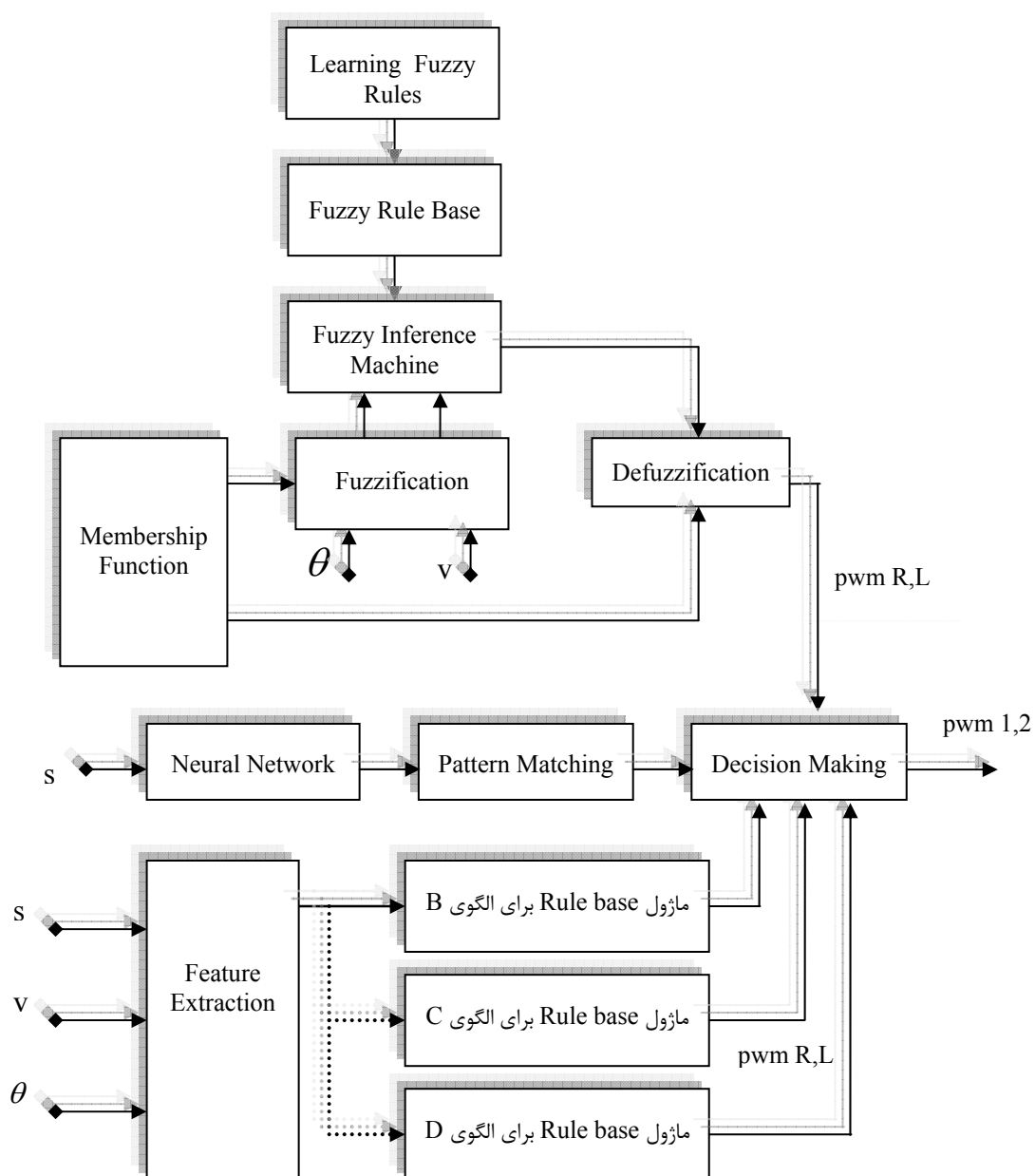
۲-۱- سیستم مکانیکی

این سیستم شامل اجزایی چون موتورها، گیربکس ها، شاسی، چرخهای عقب و جلو، بردهای الکترونیکی، اتصالات، دو ورودی pwm برای موتورهای راست و چپ و یک خروجی (جابه جایی سیستم مکانیکی) می باشد. در این مقاله بنا به اهمیت بخش سیستم هیبرید، جزئیات این بخش مورد بررسی قرار نخواهد گرفت.

۲-۲- بلاک آشکار ساز

این بخش مطابق شکل (۴)، شامل یک بلاک سنسور انعکاسی (دارای ۱۲ سنسور مادون قرمز انعکاسی که به صورت نیم دایره در جلوی ربات قرار می گیرد) می باشد. ورودی ۱۲ سنسور توسط یک مبدل ADC^۴ به عددی در بازه (۰ تا ۲۵۵) تبدیل می شود. خروجی ۱۲ سنسور انعکاسی، مستقیماً به خروجی s و نیز به بلاک آشکار ساز انحراف که جهت محاسبه انحراف لحظه ای ربات از مسیر، از یک الگوریتم ریاضی پیشنهادی استفاده می کند، وارد می شود. خروجی حاصل از این الگوریتم یک زاویه (θ) معادل انحراف ربات بر روی مسیر می باشد [۶].

بلاک آشکار ساز سرعت از دو شفت انکدر تشکیل شده است که توسط ۲ سنسور شمارنده، با دریافت دوران موتورها به یک نسبت معین، تعدادی پالس متناسب با سرعت هر یک از موتورها (v) تولید می کند. این پالس ها توسط الگوریتم سیستم به عددی در بازه (۰ تا ۸۰۰) دور در دقیقه^۵ معادل سرعت موتورها تبدیل می شود.



شکل ۶: سیستم کنترل هیبرید

رابطه (۲) و سایر توابع عضویت از نوع سیگموئید با رابطه (۳) می باشند.

$$F(x, N, c) = \exp\left(\frac{-(x-c)^2}{2N^2}\right) \quad (۲)$$

$$F(x, \alpha, c) = \frac{1}{1 + e^{-\alpha(x-c)}} \quad (۳)$$

در این کنترلر مطابق روابط (۴) و (۵) از متد استنتاج max-min استفاده شده است [۲].

$$\mu B_1^i(y_1) = \max_i[\min[\mu A_1^i(x_1), \mu A_2^i(x_2)]] \quad i = 1..3 \quad (۴)$$

$$\mu B_2^i(y_2) = \max_i[\min[\mu A_1^i(x_1), \mu A_2^i(x_2)]] \quad i = 1..3 \quad (۵)$$

جدول ۱: متغیرها و توابع عضویت قوانین فازی

x_1	متغیر سرعت (v)
x_2	متغیر انحراف (θ)
y_1	متغیر pwm1 (برای موتور راست)
y_2	متغیر pwm2 (برای موتور چپ)
μA_1^1	سرعت های کم
μA_1^2	سرعت های متوسط
μA_1^3	سرعت های زیاد
μA_2^1	انحراف ربات به چپ
μA_2^2	انحراف های کم
μA_2^3	انحراف ربات به راست
$\mu B_1^1, \mu B_2^1$	کم pwm
$\mu B_1^2, \mu B_2^2$	متوسط pwm
$\mu B_1^3, \mu B_2^3$	زیاد pwm

در غیرفازی ساز، میانگین وزن دار شده تابع عضویت، با استفاده از متد COG^{۱۴} از رابطه (۶) بدست می آید.

$$X_c = \frac{\int x \mu_A(x) dx}{\int \mu_A(x) dx} \quad (۶)$$

در نهایت مطابق شکل (۸ و ۹)، دو رویه کنترل^{۱۵} برای دو

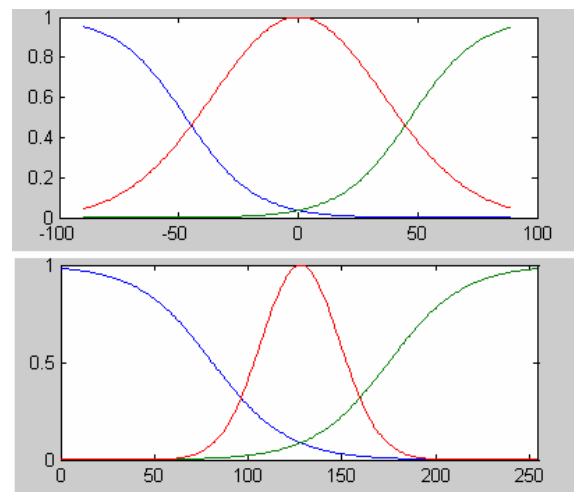
شده به موتور استنتاج^{۱۱} فازی داده می شود تا با استفاده از قوانین فازی تعریف شده، بهترین خروجی (pwm1, pwm2) توسط بلاک غیرفازی ساز^{۱۲} برای هر یک از موتورها به طور مستقل تولید شود.

در قسمت دیگر این سیستم هیبرید ورودی های s و v و θ به بلاک استخراج ویژگی^{۱۳} داده می شوند تا در هر لحظه از حرکت ربات بر روی مسیر و در هر یک از الگوهای D و C، B، که هر یک حالات خاصی را به خود اختصاص می دهند، مشخصه های مسیر استخراج شود. بلاک های مبتنی بر قانون که شامل قوانینی جهت هدایت ربات در الگوی خاص خود هستند به کمک این مشخصه ها خروجی مناسبی تولید خواهند کرد. خروجی این بلاک ها، به بلاک تصمیم گیری اعمال می شود.

۳- کنترلر فازی

این کنترلر شامل ۳ بخش فازی ساز، موتور استنتاج و غیرفازی ساز می باشد. در این مقاله از سیستم فازی ممدانی استفاده شده است. هر یک از قوانین این سیستم به صورت رابطه^۱ (۱) می باشد [۱].

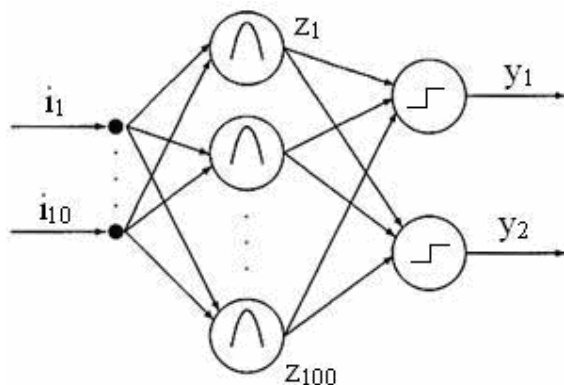
$$\text{IF } x_1 \text{ IS } A_1^i \text{ AND } x_2 \text{ IS } A_2^i \text{ THEN } y_1 \text{ IS } B_1^i \text{ AND } y_2 \text{ IS } B_2^i \quad i=1, \dots, 3 \quad (۱)$$



شکل ۷: توابع عضویت مجموعه های فازی pwm1 (پایین) و انحراف (بالا)

i تعداد قوانین می باشد. مطابق شکل (۷) توابع عضویت μA_1^2 و μA_2^2 و μB_1^2 و μB_2^2 از نوع گوسی با

های محلی، به صورت تصادفی انتخاب می شوند. مطابق شکل (۱۱)، I (ورودی s شبکه) یک بردار به طول ۱۰، دارای مقادیر نمونه برداری شده سنسورها توسط ADC می باشد. w ماتریس وزن های لایه اول یک ماتریس ۱۰۰×۱۰ و برای لایه دوم یک ماتریس ۱۰۰×۲ می باشد.



شکل ۱۱: شبکه عصبی پرسپترون چند لایه

جهت تنظیم وزن ها از رابطه (۷) استفاده می شود.

$$w_i(New) = w_i(old) + \Delta w_i \quad (7)$$

برای نورون های لایه میانی تابع آستانه^{۱۷}، تابع سیگموئید و برای نورون های لایه آخر، تابع آستانه باینری^{۱۸} مطابق رابطه (۸) انتخاب شده اند. بدین ترتیب خروجی باینری دو نورون لایه آخر به بلاک تطبیق الگو اعمال خواهد شد تا یکی از چهار الگوی مورد نظر مشخص گردد. متوسط مربعات خطای خروجی از رابطه (۹) محاسبه می شود [۲].

$$O = f(a) : \{ O = 1 \text{ if } a > 0 \text{ or } O = 0 \text{ if } a < 0 \} \quad (8)$$

$$e = \frac{1}{2}(d - o)^2 \quad (9)$$

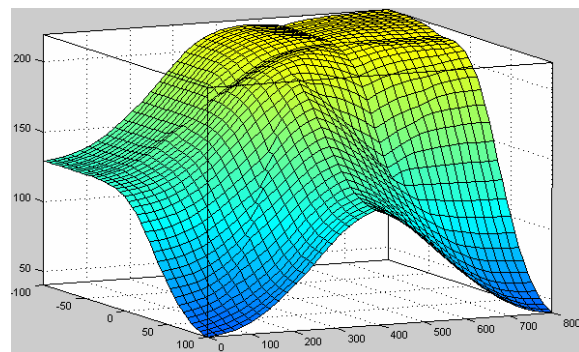
در این روابط d خروجی مطلوب شبکه و a و O به ترتیب سطح فعالیت و خروجی هر نورون می باشند. با استفاده از روابط قبل، بردار گرادیان خطا برابر رابطه (۱۰) بدست می آید [۲].

$$\nabla e = \frac{\partial e}{\partial w_i} = -(d - o)f'(w^T I)I_i \quad (10)$$

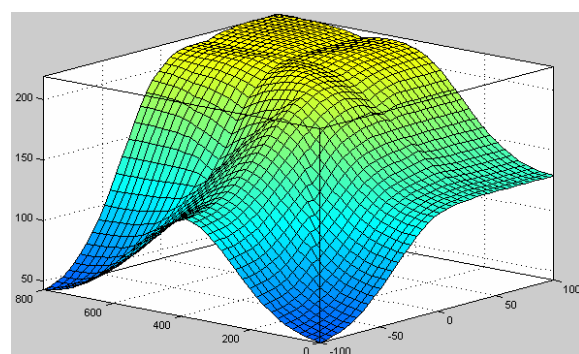
در رابطه قبل f'(.) مشتق تابع سیگموئید می باشد.

$$\Delta w = -\mu \nabla e \quad (11)$$

متغیر pwm موتور راست و چپ به طور مستقل بدست می آید.



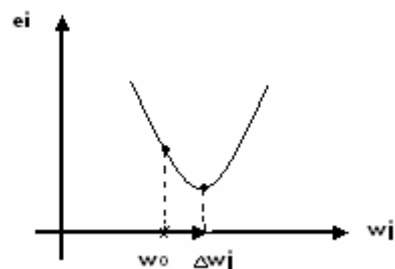
شکل ۸: خروجی pwm1 بر حسب v و theta



شکل ۹: خروجی pwm2 بر حسب v و theta

۴- شبکه عصبی

در این مقاله از شبکه عصبی پرسپترون چند لایه استفاده شده است. در این شبکه از الگوریتم پس انتشار خطا^{۱۶} برای آموزش شبکه استفاده شد. قانون دلتا مطابق شکل (۱۰) با یک روند کاملاً بهینه با محاسبه بردار گرادیان خطا، وزن های شبکه را به خوبی در خلاف جهت بردار گرادیان خطا، یعنی در خلاف جهت افزایش خطای شبکه تغییر می دهد [۲].



شکل ۱۰: تغییر وزن های شبکه در جهت مینیم شدن خطا

وزن های اولیه جهت جلوگیری از متوقف شدن در مینیمم



کنترل مسیر ربات، هشتمین کنفرانس دانشجویی و مهندسی برق ایران، ۱۳۸۴.

μ ، نرخ یادگیری، دارای مقدار یک می باشد. مقدار بیشتر آن باعث نوسان وزن ها و مقدار کمتر آن، باعث کندی همگرایی وزن ها به مقدار نهایی شان می شود. برای هر کدام از وزن های شبکه رابطه^۵ (۱۲) برقرار می باشد.

$$\Delta w_i = -\mu(d - o)f'(a)I_i \quad (12)$$

۵- نتیجه گیری

در این مقاله یک سیستم هیبرید جهت کنترل مسیر یک ربات ارائه و زیرمژول های آن نشان داده می شوند. با توجه به نتایج حاصل از شبیه سازی و مقایسه این سیستم با روش ها و الگوریتم کنترلرهای متداول در کنترل مسیر یک ربات، این نتیجه حاصل می گردد که با استفاده از یک سیستم هیبرید و به کمک متدی همچون یک کنترلر فازی با ویژگی هایی چون منطق فازی توسعه یافته، پایداری مطلوب، قابلیت تعمیم و توسعه و نتیجه گیری هایی همچون انسان با استفاده از اصطلاحات زبانی و یک شبکه عصبی با توانایی یادگیری از داده و قابلیت تعمیم قوی، حافظه مند بودن و دارا بودن عملکرد موازی، قادر به حل، ساده سازی، بهینه سازی و هوشمند سازی بسیاری از مسائل، کنترلرها، و مکانیزم ها خواهیم بود به طوری که در سایر کنترلرها بعثت فقدان این ویژگی ها نمی توان به چنین عملکرد بالایی دست یافت.

مراجع

- [1]: Nikola K. Kasabov, *Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering*, The MIT Press, Second Printing, 1998.
- [2]: Ali Zilouchian, Mo Jamshidi, *Intelligent Control Systems Using Soft Computing Methodologies*, CRC Press, 2002.
- [3]: Nikola K. Kasabov, *General Principles of Hybrid Systems*, 1995.
- [4]: Morasso, *Robot Planning in a Hybrid Connectionist System*, 1992.

[۵]: کازوتاناکا، ترجمه ع. وحیدیان کامکاد و ح. طارقیان، مقدمه ای بر منطق فازی برای کاربردهای عملی، انتشارات دانشگاه فردوسی، ۱۳۸۱.

[۶]: ع. مصدق عنبران، پیاده سازی سخت افزاری و مقایسه عملکرد کنترل فازی و روش کنترل کلاسیک پیشنهادی در

- ¹ - Hierarchical
- ² - Feed Back
- ³ - Rule Base
- ⁴ - Analog to Digital Converter
- ⁵ - Rotate Per Minute (rpm)
- ⁶ - Multilayer Perceptron
- ⁷ - Pattern Classification
- ⁸ - Pattern Matching
- ⁹ - Decision Making
- ¹⁰ - Fuzzification
- ¹¹ - Inference Machine
- ¹² - Defuzzification
- ¹³ - Feature Extraction
- ¹⁴ - Center of Gravity
- ¹⁵ - Control Surface
- ¹⁶ - Error Back Propagation
- ¹⁷ - Activation Function
- ¹⁸ - Binary Threshold Function