تولید قواعد فازی احتمالی به‌کمک یادگیری تقویتی

نعیمه محمدکریمی[۱]، فارغ‌التحصیل کارشناسی ارشد مهندسی برق و کامپیوتر، ولی درهمی[۲]، دانشیار

۱- دانشکده مهندسی برق و کامپیوتر - دانشگاه یزد- یزد- ایران- nmohammadkarimi@stu.yazd.ac.ir

۲- دانشکده مهندسی برق و کامپیوتر - دانشگاه یزد- یزد- ایران- vderhami@yazd.ac.ir

چکیده: مهم‌ترین بخش در یک سیستم فازی پایگاه قواعد آن است. یکی از مشکلات موجود در تولید قواعد فازی با داده‌های آموزشی، وجود داده‌های ناسازگار است زیرا در این‌گونه داده‌ها چند خروجی برای وضعیت‌های یکسان وجود دارد. لذا تولید قواعد و تصمیم‌گیری برای انتخاب تالی مناسب برای هر قاعده با چالش همراه خواهد بود. روش‌های موجود از برآیند حالت‌های ناسازگار استفاده می‌کنند که باعث تولید خروجی با مقدار میانگین تالی‌های مربوطه می‌شود. به‌منظور بهبود این مشکل در این مقاله از مقداردهی اولیه به‌مقدار احتمال انتخاب عمل‌ها، در یادگیری تقویتی فازی مبتنی بر معماری عملگر-نقاد استفاده می‌شود. با خوشه‌بندی داده آموزشی و استفاده از مدل سوگنوی مرتبه صفر با تعدادی عمل کاندید در هر قاعده، پارامترهای ماژول عملگر مقداردهی اولیه شده و درنهایت با معماری عملگر-نقاد و سیگنال تقویتی، به‌صورت برخط تنظیم می‌شوند. با توجه به اینکه مشکل ناسازگاری در داده‌های مربوط به ناوبری ربات نسبت به موارد دیگر نمایان‌تر است، ایده ارائه‌شده در مسئله ناوبری ربات استفاده می‌شود. آزمایش‌ها در شبیه‌ساز Webots برای ربات ایپاک انجام شده است. نتایج آزمایش‌ها حاکی از آن است که روش ارائه‌شده موجب کاهش زمان یادگیری، کاهش برخورد به موانع در مسئله ناوبری ربات با قواعد فازی کم‌تر است.

واژه‌های کلیدی: کنترل‌گر فازی، تولید قواعد فازی، داده آموزشی ناسازگار، معماری عملگر-نقاد.

Generation of Probabilistic Fuzzy Rule by Reinforcement Learning

N.MohammadKarimi[1],M.Sc, V.Derhami[2],Associate Professor

1- Faculty of Electrical and Computer Engineering, University of Yazd, Yazd, Iran, nmohammadkarimi@stu.yazd.ac.ir
2- Faculty of Electrical and Computer Engineering, University of Yazd, Yazd, Iran, vderhami@yazd.ac.ir

Abstract: Rule base is the most important part of a fuzzy inference system. Inconsistent data make some challenges in generating of fuzzy rules. In these cases, since there are multiple outputs for the same states, hence making decision for suitable consequence selection in each rule is a big challenge. Averaging of inconsistent states has been adopted by current methods and they create output with average of related consequences. The initialization of actions selection probability in fuzzy reinforcement learning based on architecture Actor-critic is used in this method. In this method, training data is clustered and zero order Sugeno method with number of candidate action in each rule are used for the initialization of the actor module parameters and they are online tuned with adopting actor-critic and reinforcement signal finally. There are many inconsistent challenges in robot navigation data in comparing other cases. Therefore the proposed method is used in robot navigation problem. The experiments are done for e-puck robot in Webots simulation. Results show that proposed method has reduced training time, collision to obstacle and fuzzy rule numbers.

Keywords: fuzzy controller, fuzzy rule generation, inconsistent data, actor-critic architecture.

تاریخ ارسال مقاله: ۱۳۹۵/۰۵/۲۷

تاریخ اصلاح مقاله: ۱۳۹۵/۰۹/۲۸

تاریخ پذیرش مقاله: ۱۳۹۵/۱۱/۱۲

نام نویسنده مسئول: ولی درهمی

نشانی نویسنده مسئول: ایران – یزد - دانشگاه یزد – ساختمان پردیس فنی و مهندسی ۱ – اتاق ۲۲۳

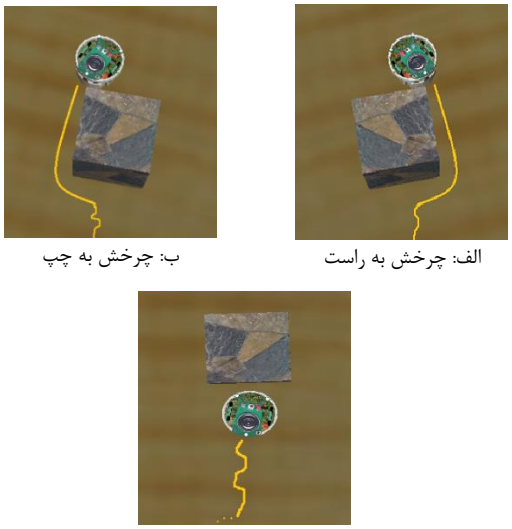
۱- مقدمه

مجموعه‌های فازی [۱] و کنترل فازی [۲] توسط دکتر لطفی‌زاده، استاد دانشگاه برکلی آمریکا، به‌ترتیب در سال ۱۹۶۵ و ۱۹۷۳معرفی گردید. سیستم‌های فازی، سیستم‌هایی دانش مبنا هستند که در آن‌ها پایگاه دانش از قواعد اگر-آنگاه فازی تشکیل شده است. اولین و مهم‌ترین قدم در ساخت این سیستم، به‌دست‌آوردن مجموعه‌ای از قواعد فازی است و روش‌های مختلفی [۳، ۴] برای تولید قواعد اگر-آنگاه فازی وجود دارد، همچنین از سیستم‌های فازی به‌عنوان تقریب‌گرهای عمومی [۵، ۶] یاد می‌شود. برای تولید قواعد فازی دو راهکار [۷] اساسی استفاده از دانش خبره و داده‌های ورودی-خروجی است. در صورت وجود دانش قبلی از سیستم، طراحی سیستم با دانش خبره انجام می‌گیرد. اگر دانش کافی از سیستم در دسترس نباشد، باید با داده‌های عددی جمع‌آوری‌شده، سیستم موردنظر طراحی گردد. در این مقاله طراحی سیستم با داده‌های عددی مورد بررسی قرار می‌گیرد.

ازجمله روش‌های تولید قواعد فازی، روش جدول جستجوی وانگ [۸] است که با هر زوج ورودی-خروجی یک قاعده تولید می‌شود لیکن امکان وجود تناقض بین داده‌ها وجود دارد. این روش با بروز قواعد با مقدم یکسان و تالی متفاوت، قاعده با بیش‌ترین درجه تحریک انتخاب می‌گردد. در روش جدول جستجوی وانگ، یک رویه سیستماتیک برای تعیین تعداد قواعد و توابع عضویت وجود ندارد و مستقل از داده‌های ورودی-خروجی است. روش‌های زیادی شبیه روش جدول جستجوی وانگ وجود دارد که از بیشینه برای انتخاب قاعده‌ای بین قواعد ناسازگار و از راهکارهای متفاوتی همچون الگوریتم ژنتیک برای تنظیم پارامترهای سیستم استفاده می‌کنند [۹]، ضعف دیگر این روش ثابت بودن تعداد قواعد است.

روش دیگر، روش حداقل مربعات بازگشتی [۸] است که از همه داده‌های ورودی-خروجی برای به‌روزرسانی پارامترهای سیستم فازی طراحی‌شده، استفاده می‌گردد. در این روش نیز راهکار سیستماتیکی برای تعیین تعداد قواعد فازی ارائه نشده است. روش دیگر خوشه‌بندی داده‌های ورودی-خروجی است که هر یک از خوشه‌ها به‌عنوان یک قاعده در نظر گرفته می‌شود. مزیت این روش، متغیر بودن تعداد قواعد فازی است. ضعف این روش، عدم راهکار مناسبی در مواجهه با داده‌های ناسازگار است زیرا هرچه خوشه‌ها کوچک می‌شوند باز مشکل ناسازگاری وجود دارد. بعضی از روش‌ها [۱۰] فقط از خوشه‌بندی داده‌های ورودی استفاده می‌کنند که بهتر است ورودی-خروجی باهم خوشه‌بندی شوند تا از دانش موجود در داده‌های خروجی نیز برای خوشه‌بندی بهره برده شود.

از دیگر روش‌هایی که با استفاده از داده‌های عددی به طراحی سیستم فازی می‌پردازد روش گروه‌بندی داده‌های ناسازگار [۱۱] است که به داده‌های ناسازگار موجود در یک گروه، احتمالی نسبت می‌دهد و درنهایت با میانگین وزن‌دار خروجی قواعد، خروجی نهایی سیستم به‌دست می‌آید. در اکثر روش‌هایی که برای بهبود مشکل ناسازگاری ارائه شده است از میانگین‌گیری استفاده می‌شود که باعث افزایش خطای سیستم می‌گردد. به‌عنوان‌مثال [۱۲] مطابق شکل ۱، در ناوبری ربات اگر ربات در موقعیتی باشد که بتواند با زاویه ۴۵ به راست (قسمت الف) و با زاویه ۴۵- به چپ (قسمت ب) بچرخد، با روش‌های موجود و میانگین‌گیری وزن‌دار خروجی قواعد، خروجی سیستم صفر که باعث حرکت ربات با زاویه صفر درجه و برخورد آن به مانع می‌گردد.



الف: چرخش به راست ب: چرخش به چپ

ج: حرکت مستقیم و برخورد به مانع

شکل۱: تأثیر سوء ناسازگاری در داده‌های آموزشی

در روشی [۱۳] دیگر برای تولید قواعد فازی از روش شبکه‌ای استفاده می‌شود لیکن تعداد قواعد فازی بیش‌تر و همچنین پیچیدگی آن هم افزایش می‌یابد. اکثر مسائل موجود در دنیای واقعی یک مسئله بهینه‌سازی با ماهیتی پویا هستند، به‌طوری‌که مقدار بهینه سراسری آن‌ها در طول زمان ممکن است تغییر کند [۱۴]، ازجمله روش‌های پویا می‌توان به روش‌های یادگیری تقویتی اشاره کرد. در مسائلی که عدم قطعیت وجود دارد بهتر است از روش‌های احتمالی استفاده شود [۱۵].

در این مقاله طراحی سیستم فازی با خوشه‌بندی، سیستم فازی سوگنو [۱۶] مرتبه صفر و برای تالی قواعد چند عمل کاندید در نظر گرفته می‌شود. برای بهبود تولید قواعد فازی با داده‌های دارای ناسازگاری از مقداردهی اولیه به مقدار احتمال انتخاب عمل‌ها، در یادگیری تقویتی فازی مبتنی بر معماری عملگر-نقاد استفاده می‌شود بدین‌صورت که از داده آموزش برای تعیین ساختار و تنظیم توابع عضویت و مقداردهی اولیه تالی قواعد سیستم کنترلی فازی استفاده شده و آنگاه مقدار نهایی تالی قواعد با عملگر معماری عملگر-نقاد و سیگنال تقویتی که به‌صورت برخط به‌دست می‌آید تنظیم می‌شود. با توجه به اینکه مشکل ناسازگاری در داده‌های مربوط به ناوبری ربات نسبت به موارد دیگر نمایان‌تر است به‌همین دلیل، ایده ارائه‌شده در مسئله ناوبری ربات استفاده شده است که حاکی از کاهش زمان یادگیری، کاهش برخورد به موانع و کاهش تعداد قواعد فازی است.

ساختار مقاله بدین شرح است. در بخش ۲ تولید قواعد فازی احتمالی و روش ارائه‌شده تشریح می‌گردد. در بخش ۳ به پیاده‌سازی و شبیه‌سازی کار پرداخته می‌شود و درنهایت بخش ۴ بحث و نتیجه‌گیری را بیان می‌کند.

۲- تولید قواعد فازی احتمالی با داده‌های عددی

روشی که در این مقاله برای بهبود تولید قواعد فازی ارائه می‌شود از دو فاز اصلی تشکیل شده است. فاز اول خوشه‌بندی داده‌های ورودی-خروجی با خوشه‌بند c-میانگین، تعیین ساختار، تنظیم پارامترهای توابع عضویت در قسمت مقدم و مقداردهی اولیه پارامترهای تالی قواعد فازی است. فاز دوم، تنظیم پارامترهای تالی قواعد به‌کمک یادگیری تقویتی مبتنی بر معماری عملگر-نقاد و تنظیم دقیق‌تر پارامتر پهنای توابع گوسین در قسمت مقدم به‌صورت برخط است.

۲-۱- طراحی سیستم فازی احتمالی

سیستم فازی سوگنو مرتبه صفر مطابق ذیل در نظر گرفته می‌شود:

$$\begin{aligned} &Rule_i: if\ x_1\ is\ A_{i1}\ and\ x_2\ is\ A_{i2}\ and\ \dots\ x_n\ is\ A_{in},\ then \\ &\quad y\ is\ action_1\ with\ probability\ p_{i1}\ and \\ &\quad y\ is\ action_2\ with\ probability\ p_{i2}\ and \\ &\quad \dots \\ &\quad y\ is\ action_m\ with\ probability\ p_{im} \end{aligned}$$

$$\sum_{j=1}^{m} p_{ij} = 1,\ \text{i=1,…,c}$$

به‌کمک خوشه‌بند c-میانگین، داده‌های ورودی-خروجی جمع‌آوری‌شده توسط ناظر به c خوشه تقسیم‌بندی می‌شوند [۱۷]. فرض کنید $s = A_i = A_{i1} \times \dots \times x_1 \times \dots \times x_n$ بردار n بعدی متغیرهای ورودی و A_{in}، n مجموعه فازی محدب اکیداً نرمال برای i امین قاعده باشد. m تعداد عمل‌های گسسته ممکن در هر قاعده و $action_{ij}$، j امین عمل کاندید برای قاعده iام و p_{ij}، ارزش احتمالی عمل jام در قاعده iام است. در هر قاعده از m عمل موجود در تالی قواعد، عملی مناسب انتخاب می‌شود و برای خروجی سیستم فازی، ترکیب وزن‌دار آن‌ها محاسبه می‌گردد.

مقدار احتمالی تالی قواعد از رابطه (۱) که نرمال‌شده تعداد داده‌های خروجی موجود در هر خوشه به نزدیک‌ترین عمل کاندید در تالی قواعد است، به‌دست می‌آید.

$$p_{ij} = \frac{\sum^{k}_{|output_k - action_j| < |output_k - action_{t\neq j}|} 1}{n'} \quad \begin{matrix} i=1:c,\ j=1:m, \\ k=1:n'\ (n'=\text{number of data in cluster}_i) \end{matrix} \qquad (۱)$$

این مقدار احتمالی، به عملگر معماری عملگر-نقاد تزریق می‌شود. با این کار درواقع دانش موجود در داده‌های ورودی-خروجی به عملگر معماری عملگر-نقاد داده می‌شود. با توجه به اینکه از سیستم فازی سوگنو مرتبه صفر استفاده شده است، خروجی سیستم فازی مطابق رابطه (۲)، جمع وزن‌دار خروجی قواعد است [۱۸].

$$\hat{y} = \frac{\sum_{i=1}^{c} w_i y_i}{\sum_{i=1}^{c} w_i} \qquad (۲)$$

که درجه تحریک هر قاعده طبق رابطه (۳) برابر است با:

$$w_i = \prod_{j=1}^{n} \mu(A_i^j) \qquad (۳)$$

و درجه تطابق داده‌های ورودی از رابطه (۴) به‌دست می‌آید.

$$\mu(A_j^i) = \exp\left(-\frac{(x_j - v_{ij})^2}{2{\sigma_{ij}}^2}\right) \quad \begin{matrix} i=1,\dots,c \\ j=1,\dots,n \end{matrix} \qquad (۴)$$

به‌ترتیب σ_{ij},v_{ij} به‌عنوان مرکز و پهنای توابع گوسین در نظر گرفته می‌شود.

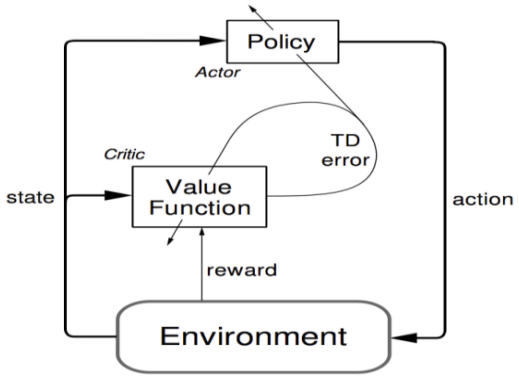
۲-۲- تنظیم دقیق پارامترهای سیستم طراحی‌شده

پارامترهای p_{ij} در طی آموزش سیستم به‌صورت برخط با سیگنال‌های دریافتی به‌روزرسانی می‌شود تا بهترین عمل از تالی قواعد بر مبنای آن‌ها انتخاب گردد [۱۹]. معماری عملگر- نقاد [۲۰، ۲۱] یکی از معروف‌ترین روش‌ها در یادگیری تقویتی پیوسته است که در مسائل بسیاری به‌کار گرفته شده است. این الگوریتم دارای دو بخش جداگانه سیاست (عملگر) و نقاد است. در شکل ۲، بخش عملگر، سیاست عمل خروجی را تولید می‌کند و بخش نقاد، تابع ارزش را تقریب می‌زند. در اکثر روش‌های ارائه‌شده، بخش نقاد تقریب‌زننده تابع ارزش حالت است.

برای بیان روابط ریاضی در این معماری، فرض کنید که از مدل فازی سوگنو مرتبه صفر با c قاعده، برای دو بخش عملگر و نقاد استفاده شده باشد، عمل نهایی a (خروجی بخش عملگر) و مقدار ارزش تقریب‌زده‌شده $p_t(A_t)$ (خروجی بخش نقاد) به‌ترتیب در روابط (۵) و (۶) محاسبه می‌شوند.

$$a_t(A_t) = \sum_{i=1}^{c} \mu_{i(A_t)}\, a_i(t) \qquad (۵)$$

$$p_t(A_t) = \sum_{i=1}^{c} \mu_{i(A_t)}\, w_i(t) \qquad (۶)$$



شکل۲: ساختار عملگر-نقاد [۲۲]

که $\mu_i(A_t)$ شدت تحریک نرمالیزشده قاعده iام برای حالت A_t، $a_i(t)$ پارامترهای وزن بخش عملگر و $w_i(t)$ پارامترهای وزن بخش نقاد هستند.

بعد از اجرای هر عمل، خطای تفاضل موقتی در رابطه (۷) محاسبه می‌گردد که r سیگنال تقویتی و γ فاکتور نزولی می‌باشند.

$$\varepsilon(t) = r_{t+1} + \gamma p_t(A_{t+1}) - p_t(A_t) \quad (۷)$$

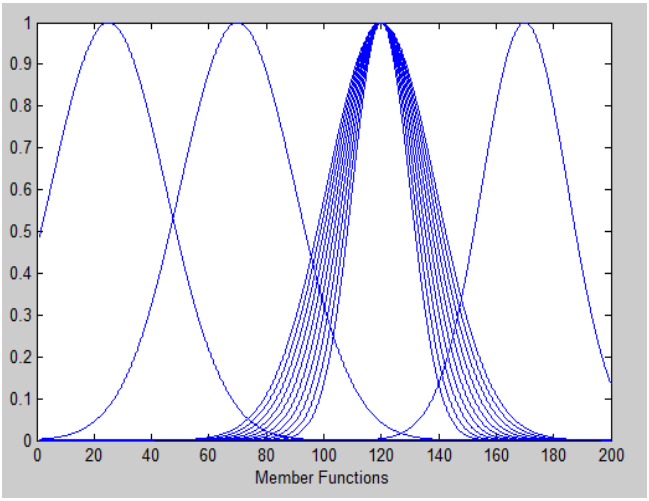
از این خطا برای به‌روزرسانی مقادیر وزن هر دو بخش نقاد و عملگر با استفاده از روش پس‌انتشار خطا، مطابق با روابط (۸) و (۹) استفاده می‌شود که α و β نرخ‌های آموزش برای دو بخش عملگر و نقاد هستند.

$$a_i(t+1) = a_i(t) + \beta_t \times \varepsilon_t \times \mu_i(A_t) \quad (۸)$$

$$w_i(t+1) = w_i(t) + \alpha_t \times \varepsilon_t \times \mu_i(A_t) \quad (۹)$$

لازم به‌ذکر است که با به‌روزرسانی پارامترهای تالی باید شرط $\sum_{j=1}^{m} p_{ij} = 1$ حفظ گردد. به این صورت که اگر عملی جریمه گرفت مقداری که از ارزش احتمالی آن کم می‌شود بین دیگر عمل‌ها تقسیم و به ارزش احتمالی آن‌ها اضافه گردد. اگر عملی پاداش گرفت باید مقداری که پاداش گرفته بین دیگر عمل‌ها تقسیم و از ارزش احتمالی آن‌ها کسر شود.

برای تولید قواعد فازی به‌کمک خوشه‌بندی در مسائلی مثل ناوبری ربات محدوده داده‌ها گسترده است و جمع‌آوری همه جانبه داده مقدور نیست. لیکن در بعضی از شرایط حالتی پیش می‌آید که برای داده ورودی، همه قواعد با درجه تحریک نزدیک به صفر، تحریک می‌شوند. این مشکل وقتی نمایان می‌شود که باید برای به‌روزرسانی ارزش‌ها در عملگر و نقاد از درجه تحریک قواعد استفاده شود. به‌منظور رفع این مشکل، وقتی درجه تحریک تمام قواعد برای داده ورودی نزدیک به صفر شود، باید پهنای توابع گوسین در قسمت مقدم قواعد به‌روزرسانی گردد. لیکن چهار قاعده از قواعدی که درجه تحریک بیشتری دارند، انتخاب می‌شود، با توجه به اینکه And-method استفاده‌شده کمینه است لذا کوچک‌ترین درجه تطابق بین ابعاد مختلف هر یک از چهار قاعده انتخابی را در نظر گرفته و پهنای تابع گوسین آن به‌روزرسانی می‌شود. به این صورت که ۰/۰۱ به پهنای تابع گوسین مربوطه اضافه می‌گردد (شکل ۳) و این‌کار تکرار می‌شود تا حداقل درجه تحریک یک قاعده از کل قواعد فازی، به حداقل مقدار ۰/۰۰۰۱ برسد [۲۳].



شکل ۳: توابع عضویت به‌روزرسانی شده

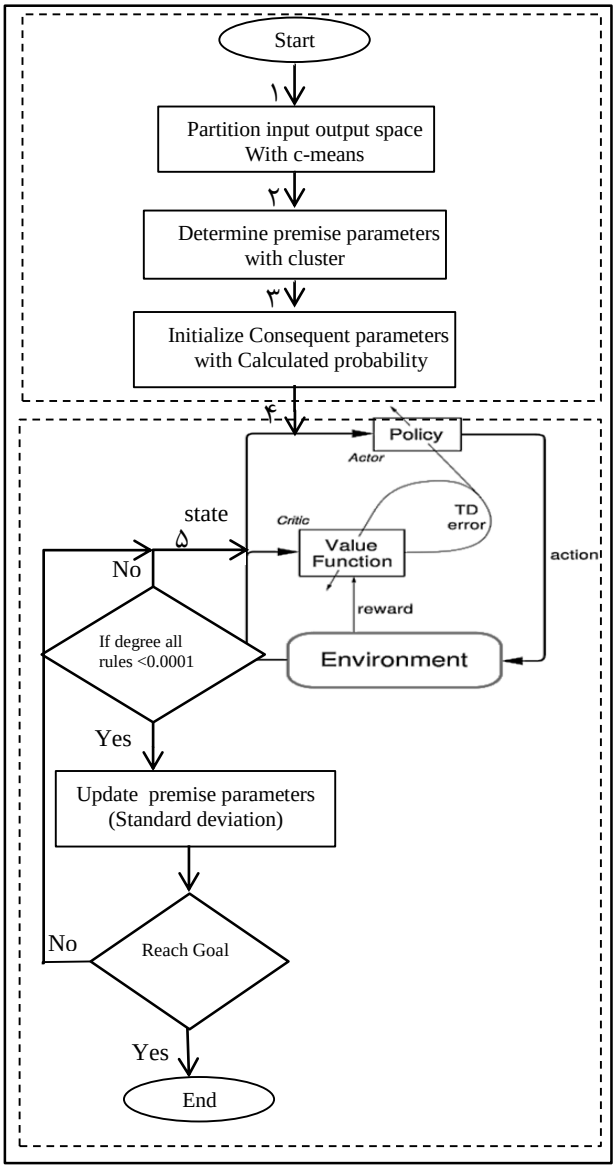
شبه‌کد به‌روزرسانی برخط پهنای توابع گوسین در رابطه (۱۰) آمده است.

While **firing strength of all rule** <0.0001
Select four **rules** that have highest firing strength(r1,r2,r3,r4)
$\sigma_{[k1,k2,k3,k4]}(n') = \sigma_{[k1,k2,k3,k4]}(n') + 0.01,$
(n'=min(degree of **membership**(r1,r2,r3,r4)) (۱۰)
End

روش ارائه‌شده را یادگیری تقویتی فازی-عملگر-نقاد با ناظر[۱] می‌نامیم که مراحل آن به‌طور خلاصه در ذیل آمده است:

۱. خوشه‌بندی داده‌های ورودی-خروجی جمع‌آوری‌شده توسط ناظر با الگوریتم c-میانگین
۲. تنظیم توابع عضویت مقدم قواعد فازی
۳. مقداردهی اولیه احتمالی پارامترهای تالی قواعد
۴. تنظیم نهایی پارامترهای تالی قواعد فازی به‌کمک معماری عملگر-نقاد
۵. تنظیم دقیق‌تر پهنای توابع گوسین در قسمت مقدم قواعد به‌صورت برخط

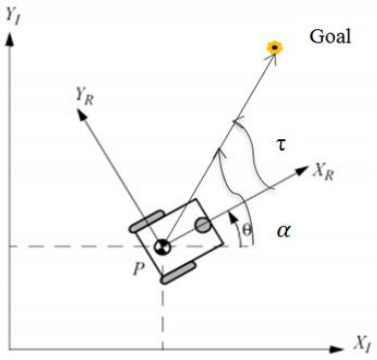
در فلوچارت شکل۴ مراحل کار به‌خوبی نشان داده شده است.



شکل۴: روش ارائه‌شده

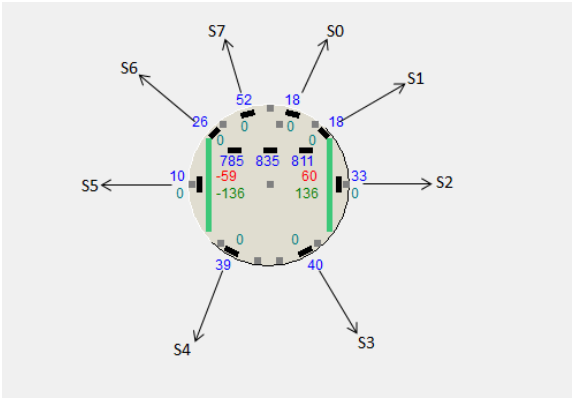
۳- شبیه‌سازی

به‌منظور پیاده‌سازی روش ارائه‌شده، از شبیه‌ساز Webots [۲۴] برای ربات ایپاک [۲۵] استفاده شده است. در ابتدا باید محیط شبیه‌سازی، موانع و ربات مهیا و نقاط شروع حرکت ربات و هدف آن مشخص گردد. شبیه‌ساز Webots یک قسمت گرافیکی و یک قسمت کدنویسی دارد. از قسمت گرافیکی برای نشان‌دادن محیط و موانع استفاده شده است و قسمت کدنویسی آن به زبان C است. این محیط شبیه‌ساز، صفحه‌ای برای شبیه‌سازی به ابعاد ۹۰۰×۹۰۰ میلی‌مترمربع دارد. موانع در جاهای فرضی قرار داده شده تا آموزش انجام شود. برای کار با ربات ایپاک در این محیط نیاز به اطلاعات سنسورها و زاویه پیشانی ربات با هدف (τ، مطابق شکل ۵) است.



شکل۵: زاویه پیشانی ربات با هدف [۲۶]

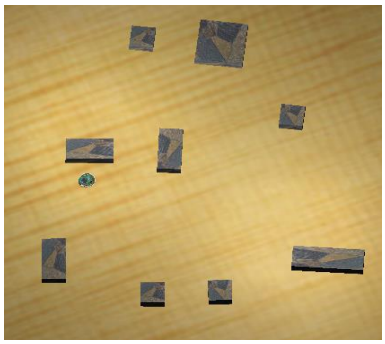
نحوه چیدمان سنسورهای ربات در شکل۶ نشان داده شده است. برای کار با ربات از سنسورهای صفر تا ۲ و ۵ تا ۷ استفاده شده است. سنسورهای صفر و ۷ برای کنترل جلوی ربات، از سنسورهای ۱و۲ برای کنترل سمت راست و سنسورهای ۵ و ۶ برای کنترل سمت چپ استفاده شده است. کنترل‌گر دارای چهار ورودی (ورودی اول تا سوم به‌ترتیب فاصله ربات با مانع از جلو، راست و چپ است و ورودی چهارم زاویه پیشانی ربات با هدف) و یک خروجی (زاویه چرخش پیشانی ربات در نزدیکی به مانع) است.



شکل۶: نمایی از سنسورهای ربات

سنسورها عددی بین ۰ تا ۳۰۰۰ را نشان می‌دهند، عدد ۰ تا ۱۰۰ موقعیتی را نشان می‌دهد که ربات هیچ مانعی را حس نمی‌کند و ۳۰۰۰ هنگامی است که ربات به مانع چسبیده باشد. زاویه پیشانی ربات با هدف عددی بین -180° و 180° است. خروجی کنترل‌گر، زاویه چرخش پیشانی ربات، عددی بین $+45^\circ$ و -45° در نظر گرفته شده است. برای هر چرخ ربات یک رمزگذار[۲] تعبیه شده است که با شمارش پالس‌های حاصل از آن می‌توان مسافت پیموده‌شده توسط هر چرخ را محاسبه کرد [۲۵]. در شکل۷ نمونه‌ای از محیط شبیه‌ساز در مرحله آموزش نشان داده شده است.

هر رویداد[۳] در بخش آموزش شامل شروع از مبدأ و رسیدن به هدف است.



شکل۷: نمایی از محیط آموزش

موقعیت هدف و شروع حرکت ربات در هر رویداد متفاوت است. شرط توقف در بخش آموزش این است که ربات با ۱۰ اجرای متوالی، بدون شکست به هدف برسد وگرنه باید با ۵۰۰ جفت موقعیت تصادفی برای ربات و هدف، در محیط آموزش داده شود. بعد از آموزش ربات وارد مرحله آزمایش می‌شود. شماره رویدادها در پایان آموزش به‌عنوان معیار زمان آموزش[۴] (LDI) در نظر گرفته می‌شود.

در این مقاله عامل باید یاد بگیرد از نقطه شروع حرکت کرده و بدون برخورد به موانع، به هدف برسد لیکن ابتدا عامل آموزش داده می‌شود. بدین منظور توسط ناظر ۲۷۹۱ داده با صفحه کلید و شبیه‌ساز جمع‌آوری شده است. به‌کمک این داده‌ها ساختار سیستم فازی موردنظر و پارامترهای مقدم تنظیم و پارامترهای تالی مقداردهی اولیه می‌شوند. در مرحله دوم آموزش از روش SAC-RL برای تنظیم پارامترهای تالی استفاده می‌شود.

برای محاسبه میزان جریمه و پاداشی که در هر وضعیت، سیستم دریافت می‌کند، به دو معیار فاصله ربات از مانع و زاویه پیشانی ربات با هدف نیاز است. معیار فاصله طبق رابطه (۱۱) تعریف می‌شود.

$$d = \min(d_{face})\ face\epsilon\{right, left, front\} \qquad (۱۱)$$

اگر d برابر صفر شود یک شکست[۵] (برخورد ربات به مانع) تلقی می‌شود و سیگنال تقویتی دریافتی ۱- است. هرگاه فاصله مرکز ربات تا هدف ۲۰ میلی‌متر باشد نشان‌دهنده رسیدن ربات به هدف است و سیگنال تقویتی دریافتی برابر ۱ است. در غیر این صورت طبق رابطه (۱۲) برحسب زاویه پیشانی ربات و میزان نزدیکی به مانع، جایزه یا جریمه‌ای بین ۱و ۱- دریافت می‌کند [۱۲].

$$\begin{cases} -1 & failure \\ -0.5 & d < 0.075 \\ \dfrac{\Delta}{150} & \Delta > 0 \ \&\ \ d \geq 0.07 \\ -0.01 + \dfrac{\Delta}{150} & \Delta \leq 0 \ \&\ \ d \geq 0.075 \\ 1 & goal \end{cases} \qquad (۱۲)$$

$$\Delta = |\theta(t-1)| - |\theta(t)|$$

۵ اجرای مستقل صورت گرفته است که هر اجرا شامل دو مرحله آموزش و آزمایش است. بعد از آموزش، برای بررسی صحت عملکرد روش ارائه‌شده، سیستم وارد مرحله آزمایش می‌شود. در مرحله آزمایش سیستم، ۱۰ محیط در نظر گرفته شده است که در ۸ محیط اول تنها موقعیت شروع و هدف تغییر می‌کند. شکل ۸ محیط‌های آزمایش را نشان می‌دهد.

شکل۸: محیط‌های آزمایش

کیفیت عملکرد سیستم طراحی‌شده در محیط آزمایش با معیارهای تعداد برخورد به موانع و مسافت طی‌شده تا رسیدن به هدف ارزیابی می‌شود. برای بررسی صحت عملکرد روش ارائه‌شده، دو سیستم فازی متفاوت دیگر نیز طراحی شده و سه سیستم باهم مقایسه می‌شوند. تالی هر قاعده در سه سیستم فازی طراحی‌شده با ۱۳ عمل کاندید به‌صورت {۴۵، ۳۰، ۲۰، ۱۵،۱۰ ، ۵، ۰،۵-، ۱۰-، ۱۵-، ۲۰-، ۳۰-، ۴۵-} تعریف شده است.

روش ارائه‌شده (SAC-RL1) با روش عملگر-نقاد(AC) [۲۰] که تولید قواعد فازی با تقسیم‌بندی شبکه‌ای و به‌ترتیب ۲و۳،۲،۲ مجموعه فازی برای ورودی‌های کنترل‌گر در نظر گرفته شده است و هیچ دانش اولیه‌ای به عملگر تزریق نمی‌شود، مقایسه می‌گردد. با سیستم فازی دیگری که با خوشه‌بندی، قواعد آن تنظیم شده است و هیچ دانش اولیه‌ای به عملگر اعمال نمی‌شود (SAC-RL2) و همچنین با روش SFSL [۱۲] که تولید قواعد فازی با تقسیم‌بندی شبکه‌ای و به‌ترتیب ۲و۲،۳،۲ مجموعه فازی برای ورودی‌های کنترل‌گر در نظر گرفته شده است، مقایسه صورت می‌گردد.

نتایج شبیه‌سازی در جدول ۱ آورده شده است. ستون دوم از چپ LDI که متوسط‌گیری از ۵ اجرای مختلف است و سومین ستون متوسط تعداد برخورد به موانع در مرحله آموزش و ستون چهارم و پنجم به‌ترتیب متوسط مسافت طی‌شده و تعداد برخورد به موانع در مرحله آزمایش است.

جدول ۱: نتایج شبیه‌سازی در مسئله ناوبری ربات

method	Ave. LDI	Failure rate1	Ave. Distance	Failure rate 2
SAC-RL1	۱۷	۱۰	۷۷	صفر
AC	۲۷	۲۱۴	۷۷	صفر
SAC-RL2	۲۰	۴۲	۷۹	۱
SFSL	۴۰	۳۰	۷۸	۹

از نتایج مشهود است روش SAC-RL1 از نظر تعداد برخورد به موانع بهبود چشم‌گیری داشته است. درواقع ۹۵٪ بهتر از روش AC و ۷۶٪ بهتر از روش SAC-RL2 و همچنین ۶۷٪ بهتر از روش SFSL است. لازم به ذکر است که روش SAC-RL2 از روش AC ، ۸۰٪

برخورد کم‌تری به مانع داشته است. از این مقایسه می‌توان نتیجه گرفت که با قواعد کم‌تر و کاراتر حتی بدون تزریق دانش اولیه به عملگر معماری عملگر-نقاد نتایج قابل‌توجهی حاصل می‌گردد اما SFSL نسبت به SAC-RL2 ۲۹٪ برخورد کم‌تری داشته است که این به‌خاطر تزریق دانش اولیه است. به‌طور متوسط، روش SAC-RL1 ۸۰٪ بهتر عمل می‌کند. با توجه به اینکه هدف بهبود تولید قواعد فازی با استفاده از داده‌های دارای ناسازگاری است تعداد برخورد به موانع، اصلی‌ترین معیار مقایسه است.

مقایسه دیگر با معیار Ave.LDI است، روش SAC-RL1 از روش AC ۳۷٪ و از روش SAC-RL2 ۱۵٪ و از روش SFSL ۵۸٪ سریع‌تر کنترل‌گر فازی را تنظیم کرده است. به‌عبارت‌دیگر سرعت آموزش به‌طور متوسط ۳۷٪ افزایش یافته است. مقایسه دیگر مسافت طی‌شده در مرحله آزمایش است چون هر چهار روش در تمام تکرارها در مرحله آزمایش به هدف رسیده‌اند مسافت طی‌شده تا هدف برای آن‌ها تقریباً برابر است. از نقطه نظر تعداد برخورد به مانع در مرحله آزمایش، چون هر چهار روش در مرحله یادگیری، به‌خوبی آموزش دیده‌اند، در مرحله آزمایش نیز به‌خوبی عمل کرده‌اند. روش SAC-RL2 یک برخورد داشته است که البته رقم قابل‌توجهی نیست. این یک برخورد به این دلیل رخ داده است که چون از خوشه‌بندی در این روش استفاده شده و در مرحله جمع‌آوری‌داده نمی‌توان همه داده‌ها را جمع‌آوری کرد. هر چند در مرحله آموزش پهنای توابع گوسین به‌صورت برخط، تنظیم می‌شود و این مشکل تا حد زیادی حل شده است ولی با به‌وجودآمدن شرایط غیرمنتظره ربات یک برخورد داشته است وگرنه در کل روش SAC-RL2 از روش AC به‌طور قابل‌توجهی بهتر عمل کرده است. روش SFSL در مرحله آزمایش ۹ برخورد داشته است که می‌توان نتیجه گرفت روش‌های تولید قواعد فازی با خوشه‌بندی و با تزریق دانش اولیه نسبت به روش‌های تولید قواعد فازی با تقسیم‌بندی شبکه‌ای و داشتن تزریق دانش اولیه بهتر عمل می‌کنند.

در جدول ۲ مقایسه‌ای بین این چهار روش از نظر تعداد قواعد فازی (پیچیدگی سیستم) یا به‌عبارت‌دیگر تعداد خوشه‌ها صورت گرفته است که نشانگر پیچیدگی کم‌تر سیستم طراحی‌شده با خوشه‌بندی است. تعداد قواعد در طراحی سیستم با خوشه‌بندی به نصف تعداد قواعد در طراحی سیستم با تقسیم‌بندی شبکه‌ای کاهش یافته است به‌عبارت‌دیگر پیچیدگی سیستم با خوشه‌بندی ۵۰٪ کم‌تر از طراحی سیستم با تقسیم‌بندی شبکه‌ای است.

جدول۲: مقایسه بین سه روش از نظر تعداد قواعد فازی (تعداد خوشه‌ها)

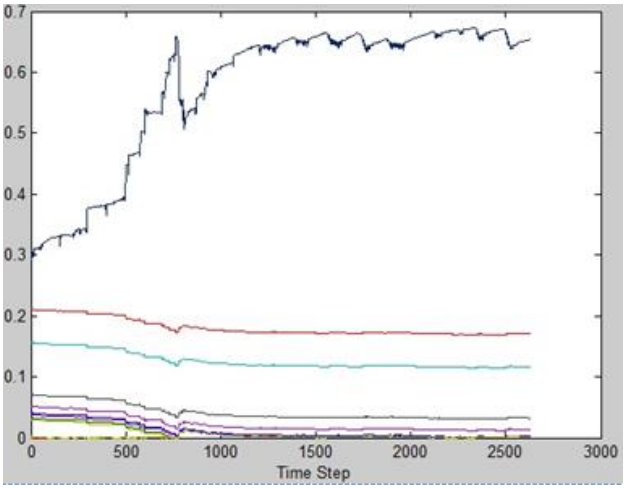
روش	تعداد قواعد فازی
SAC-RL1	۱۲
AC	۲۴
SAC-RL2	۱۲
SFSL	۲۴

در جدول ۳ واریانس نتایج چهار روش مختلف بیان شده است.

جدول ۳: واریانس نتایج شبیه‌سازی در مسئله ناوبری ربات

method	Var LDI	Var. Failure rate1	Var. Distance	Var. Failure rate2
SAC-RL1	۳/۸	۷۱/۳	۱۰/۹۶	۰
AC	۱۳۷/۷	۸۰۶۲	۰/۰۲۳	۰
SAC-RL2	۲۳/۲	۳۸۸/۷	۳/۳۵	۱
SFSL	۴۹۴/۰۴	۳۰۶/۹۳	۱۲/۵۵	۲/۰۱

در شکل ۹ نحوه تغییرات ارزش احتمالی عمل‌های کاندید در تالی قاعده دوم را نشان می‌دهد. همان‌طور که مشخص است ارزش احتمالی مربوط به عمل سوم (-۲۰°) از همان ابتدای کار به‌عنوان عملی برتر است و با افزایش ارزش احتمالی به‌عنوان عمل انتخابی قاعده مربوطه شناخته شده است. تغییرات ارزش دیگر عمل‌ها بعد از مدت کوتاهی تقریباً ثابت می‌شود. همان‌طور که در شکل مشخص شده است حدوداً در گام ۷۷۵ تا ۷۸۰ ربات ۵ برخورد به مانع داشته است و با دریافت جریمه نمودار ارزش عمل نزول پیدا کرده است که با دریافت پاداش‌های بعدی دوباره صعود می‌کند.



شکل۹: نمودار تغییرات ارزش احتمالی عمل‌ها در تالی قاعده دوم

۴- بحث و نتیجه‌گیری

ابتدا داده‌های جمع‌آوری‌شده به c خوشه، دسته‌بندی شدند و به‌کمک یادگیری با ناظر ،توابع عضویت و پارامترهای مقدم، تنظیم و پارامترهای تالی قواعد با احتمال مقداردهی اولیه شدند. سپس این احتمال به عملگر معماری عملگر-نقاد تزریق گردید و با کمک یادگیری تقویتی به تنظیم دقیق‌تر پارامترهای تالی پرداخته شد. همچنین در حین تنظیم پارامترهای تالی با یادگیری تقویتی، تنظیم دقیق‌تر پارامتر پهنای تابع گوسین به‌صورت برخط صورت گرفت. چون ناسازگاری در ناوبری ربات خیلی نمایان است از شبیه‌ساز Webots برای شبیه‌سازی اپیاک در این مسئله استفاده شد. نتایج، حاکی از بهتربودن مقداردهی احتمالی عملگر در معماری عملگر-نقاد نسبت به

روش‌هایی که هیچ دانش اولیه‌ای تزریق نمی‌شود، دارد. این روش در تعداد برخورد به مانع، زمان آموزش و تعداد قواعد برای کنترل سیستم (پیچیدگی‌کم‌تر سیستم)، بهبود خوبی داشته است.

مراجع

[1] L. A. Zadeh, "Fuzzy sets," *Information And Control*, vol. 8, pp. 338-353, 1965.

[2] L. A. ZADEH, "Outline of a new approach to the analysis of complex systems and decision processes," *IEEE Transactions On Systems, Man, And Cybernetics*, vol. 3, no. 1, pp. 28-44, 1973.

[3] L.-X. Wang and J. M. Mendel, "Generating fuzzy rules by learning from examples," *IEEE Transactions On Systems, Man, And Cybernetics*, vol. 22, no. 6, p. 1414–1427, 1992.

[4] X. Wang, X. Liu, W. Pedrycz and L. Zhang, "Fuzzy rule based decision trees," *Pattern Recognition*, vol. 48, no. 1, pp. 50-59, 2015.

[5] C. Li, J. Zhou, Q. Li and X. An, "A new T–S fuzzy-modeling approach to identify a boiler–turbine system," *Expert Systems with Applications*, vol. 37, no. 3, p. 2214–2221, 2010.

[6] X.-J. Zeng and M. G. Singh, "Approximation accuracy analysis of fuzzy systems as function approximators," *IEEE Transactions on Fuzzy Systems*, vol. 4, no. 1, p. 44–63, 1996.

[7] R. BABUSKA, Fuzzy and neural control, 2004.

[8] L.-X. Wang, A cource fuzzy system and control, Prentice Hall, 1997.

[9] D. Meng and Z. Pei, "Extracting linguistic rules from data sets using fuzzy logic and genetic algorithms," *Neurocomputing*, vol. 78, no. 1, pp. 48-54, 2012.

[10] K.-J. Park and D.-Y. Lee, "Evolutionary design of fuzzy inference systems by means of fuzzy partition of input space," *International Journal of Software Engineering and Its Applications*, vol. 7, no. 2, pp. 113-124, 2013.

[11] M. Tang, X. Chen, W. Hu and W. Yu, "Generation of a probabilistic fuzzy rule base by learning from examples," *Information Sciences*, vol. 217, p. 21–30, 2012.

[12] F. Fathinezhad and V. Derhami, "A novel supervised fuzzy reinforcement learning for robot navigation," *Journal of Control*, vol. 6, no. 3, pp. 1-10, 2012.

[13] F. Fathinezhad, V. Derhami and M. Rezaeian, "Supervised fuzzy reinforcement learning for robot navigation," *Applied Soft Computing*, vol. 40, no. 3, pp. 33-41, 2016.

[14] M. Mohammadpour and H. Parvin, "Chaotic genetic algorithm based on clustering and memory for solving dynamic optimization problems," *Tabriz Journal of Electrical Engineering*, vol. 46, no. 3, pp. 299-318, 2016.

[15] A. Saberi Noughabi, H. Badrsimaei and M. Farshad, "A Probabilistic Method to Determine the Optimal Setting of Combined Overcurrent Relays considering Uncertainties," *Tabriz Journal of Electrical Eng*, vol. 47, no. 1, 2017.

[16] T. TAKAGI and M. SUGENO, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Transactions On Systems, Man, And Cybernetics*, vol. 15, no. 1, pp. 116-132, 1985.

[17] T. M. Nguyen and Q. M. J. Wu, "Online feature selection based on fuzzy clustering and its applications," *IEEE Transactions on Fuzzy Systems*, vol. 24, no. 6, pp. 1294-1306, 2015.

[18] J.-S. R. Jang, C.-T. Sun and E. Mizutani, Neuro-fuzzy and soft computing, Prentice, 1997.

[19] V. Derhami, V. Johari Majd and M. Nili Ah, "Fuzzy sarsa learning and the proof of existence of its stationary points," *Asian Journal of Control*, vol. 10, no. 5, pp. 535-549, 2008.

[20] L. Jouffe, "Fuzzy inference system learning by reinforcement methods," *IEEE Transactions On Systems, Man, And Cybernetics*, vol. 28, no. 3, pp. 338-355, 1998.

[21] N. H. C. Yung and C. Ye, "An intelligent mobile vehicle navigator based on fuzzy logic and reinforcement learning," *IEEE Transactions On Systems, Man, And Cybernetics—Part B*, vol. 29, no. 2, pp. 314-321, 1999.

[22] R. S. Sutton and A. G. Barto, Reinforcement learning: An introduction, MIT Press Cambridge, 1998.

[۲۳] نعیمه. محمدکریمی و ولی. درهمی، "بررسی و بهبود تولید قواعد فازی و تنظیم تالی قواعد با یادگیری تقویتی،" در *دومین همایش ملی مهندسی برق ایران، بندر گز، دانشگاه آزاد اسلامی واحد بندر گز*، ۱۳۹۳.

[24] O. Michel, "Webots: professional mobile robot simulation," *Inernational Journal of Advanced Robotic Systems*, vol. 1, no. 1, pp. 40-43, 2004.

[25] F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Cianci, A. Klaptocz, S. Magnenat, J.-C. Zufferey, D. Floreano and A. Martinoli, "The e-puck, a robot designed for education in engineering," in *Proceedings of the 9th Conference on Autonomous Robot Systems and Competition*, 2009.

[26] A. Gorji Daronkolaei, V. Nazari, M. B. Menhaj and S. Shiry, "A Joint Probability Data Association Filter Algorithm for Multiple Robot Tracking Problems," *Tools in Artificial Intelligence* , no. 24, pp. 163-186, 2014.

زیرنویس‌ها

[1] Supervised Actor-Critic based Reinforcement learning(SAC-RL)

[2] Encoder

[3] Episode

[4] Learning Duration Index

[5] Failure