مجله کنترل
ISSN 2008-8345
جلد ۶، شماره ۳، پاییز ۱۳۹۱، صفحه ۱۰-۱

روشی نوین برای یادگیری تقویتی فازیِ باناظر برای ناوبری ربات

فاطمه فتحی نژاد [۱]، ولی درهمی [۲]

[۱] فارغ‌التحصیل کارشناسی ارشد مهندسی برق و کامپیوتر، گروه کامپیوتر،دانشگاه یزد fateme.fathinezhad@stu.yazduni.ac.ir

[۲] استادیار، دانشکدهٔ مهندسی برق و کامپیوتر، گروه کامپیوتر، دانشگاه یزد vderhami@yazduni.ac.ir

(تاریخ دریافت مقاله ۱۳۹۱/۴/۱۳، تاریخ پذیرش مقاله ۱۳۹۱/۷/۲)

چکیده: استفاده از یادگیری باناظر در ناوبری ربات‌های متحرک، با چالش‌های جدی از قبیل ناسازگاری و اختلال در داده‌ها، مشکل جمع‌آوری نمودن داده آموزش و خطای زیاد در داده‌های آموزشی مواجه می‌باشد. قابلیت‌های یادگیری تقویتی همچون عدم نیاز به داده آموزشی و آموزش تنها با استفاده از یک معیار اسکالر راندمان باعث کاربرد آن در ناوبری ربات شده است. از طرفی یادگیری تقویتی زمانبر بوده و دارای نرخ شکست‌های بالا در مرحله آموزش می‌باشد. در این مقاله، یک ایده جدید برای استفاده مؤثّر از هر دو الگوریتم یادگیری فوق ارائه می‌شود. یک کنترلگر فازی سوگنو مرتبه صفر با تعدادی عمل کاندید برای هر قاعده جهت تولید فرمان‌های کنترل ربات در نظر گرفته شده است. هدف از آموزش تعیین عمل مناسب برای هر قاعده است. روش ترکیبی پیشنهاد شده دو مرحله دارد. در مرحله اول، داده آموزشی با حرکت ربات توسط ناظر در محیط جمع آوری می‌شود. سپس با بهره‌گیری از روش جدید ارائه شده، پارامترهای ارزشِ هر عمل کاندید در قواعد فازی با کمک داده‌های آموزشی مقدار دهی اولیه می‌شوند. در مرحله دوم از الگوریتم سارسای فازی برای تنظیم دقیق‌تر پارامترهای تالی کنترلگر بصورت برخط استفاده می‌شود. نتایج شبیه سازی در شبیه‌ساز KiKS برای ربات خپرا حاکی از بهبود قابل توجه در زمان یادگیری، تعداد شکست‌ها، و کیفیت حرکت ربات می‌باشد.

کلمات کلیدی: ناوبری ربات ،یادگیری باناظر، یادگیری تقویتی ، کنترلگر فازی.

A Novel Supervised Fuzzy Reinforcement Learning for Robot Navigation

Fateme Fathinezhad, Vali Derhami

Abstract: Applying supervised learning in robot navigation encounters serious challenges such as inconsistence and noisy data, difficulty to gathering training data, and high error in training data. Reinforcement Learning (RL) capabilities such as lack of need to training data, training using only a scalar evaluation of efficiency and high degree of exploration have encourage researcher to use it in robot navigation problem. However, RL algorithms are time consuming also have high failure rate in the training phase. Here, a novel idea for utilizing advantages of both above supervised and reinforcement learning algorithms is proposed. A zero order Takagi-Sugeno (T-S) fuzzy controller with some candidate actions for each rule is considered as robot controller. The aim of training is to find appropriate action for each rule. This structure is compatible with Fuzzy Sarsa Learning (FSL) which is used as a continuous RL algorithm. In the first step, the robot is moved in the environment by a supervisor and the training data is gathered. As a hard tuning, the training data is used for initializing the value of each candidate action in the fuzzy rules. Afterwards, FSL fine-tunes the parameters of conclusion parts of the fuzzy controller online. The simulation results in KiKS simulator show that the proposed approach significantly improves the learning time, the number of failures, and the quality of the robot motion.

Keywords: Robot navigation, Supervised learning, Reinforcement learning, Fuzzy controller.

۱- مقدمه

ناوبری برای ربات‌های متحرک عبارت است از حرکت از یک نقطه معین و رسیدن به یک هدف مشخص در حالی‌که ربات بتواند از برخورد به موانع اجتناب کند[۲،۱]. بطور کلی در یک محیط پویا، استفاده از الگوریتم‌های سراسری[1] برای یافتن مسیر حرکت ربات، غیرممکن و یا بسیار پرهزینه است، زیرا در این روش‌ها مدل ریاضی یا نقشه کاملی از محیط مورد نیاز است. لذا چنانچه مشخصات محیط ناشناخته باشد و یا محیط در حال تغییر باشد، مسیریابی با استفاده از روش‌های طراحی مسیر محلی انجام می‌شود. روش‌های طراحی مسیر محلی از اطلاعات فراهم شده توسط حسگرهایی مانند حسگرهای سونار و یا حسگرهای مادون قرمز بهره می‌برند [۴،۳]. در میان روش‌های ارائه شده الگوریتم‌های هندسی فرض می‌کنند که حسگرهای سنجش فاصلهِ نصب شده بر روی ربات قادرند، بطور کامل موانع را بصورت برخط تشخیص دهند. این دو فرض در محیط‌های واقعی غیرقابل قبول و برآوردن آنها وقت‌گیر است [۱]. از دیگر روش‌ها، روش‌های مبتنی بر پتانسیل می‌باشند که مؤثرتر از الگوریتم‌های هندسی به نظر می‌رسند، چرا که به جزئیات کمتری از موقعیت موانع نیاز دارند، لیکن این روش‌ها نیز نقاط ضعف زیر را دارند [۵،۱]:

۱- رخداد کمینه محلی منجر می‌گردد که ربات در حلقه ایجاد شده بین موانع به دام بیفتد.

۲- حرکت ناپایدار ربات در کاربردهای عملی.

۳- مشکل پیدا کردن ضرایب مؤثر مربوط به سرعت و نیرو در محیط‌هایی با موانع درهم که ارائه مدل ریاضی برای محیط را دشوار می‌کند.

۴- افت راندمان به خاطر نایقینی و خطای مربوط به حسگرها (که برای هر حسگر متفاوت از بقیه است) به علت عملکرد بر اساس یک استراتژی از پیش تعیین‌شده و عدم وجود یادگیری.

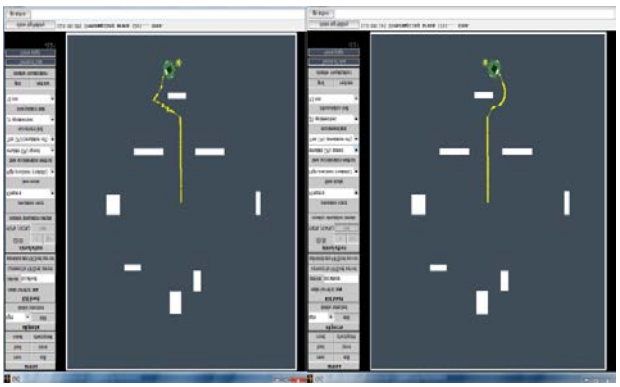
مشکلات اشاره شده در بالا، محققین را تشویق به استفاده از الگوریتم‌های یادگیرنده با استفاده از اطلاعات محلی در مسأله ناوبری ربات نموده است .این یادگیری با استفاده از اطلاعات حاصله از حسگرهای قرار گرفته بر روی ربات انجام می‌پذیرد [۶].

یادگیری باناظر یکی از روش‌های قدیمی استفاده شده برای تنظیم پارامترهای کنترلگر می‌باشد که برای آموزش کنترلگر ربات نیز از آن استفاده شده است. در این روش ابتدا ربات در محیط توسط ناظر حرکت داده شده و سپس با توجه به داده‌های جمع‌آوری شده، با استفاده از روشهای مبتنی بر گرادیان [۷] پارامترهای کنترلگر در جهت کاهش مجموع مربعات خطای خروجی تنظیم می‌شوند.

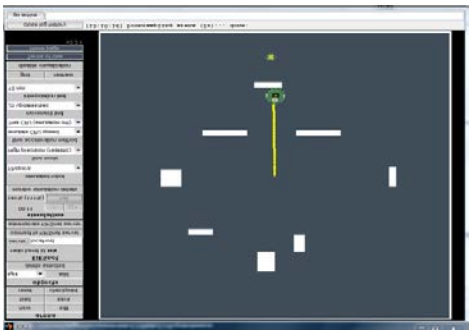
این الگوریتم در مسأله ناوبری ربات با ضعف‌های جدی مانند موارد زیر مواجه است:

۱- عدم اطلاع ناظر از فرمان کنترلی مناسب در بعضی از وضعیت‌ها: لذا در وضعیت‌های خاص خروجی تعیین شده توسط ناظر دارای خطای زیاد می باشد.

۲- ناسازگاری در داده ها: بعنوان مثال در نظر بگیرید یک ربات در جلوی مانع قرار دارد و سمت چپ و راست آن خالی می‌باشد. ناظر ممکن است در چنین وضعیتی یکبار با اعمال فرمان چرخش به راست (مثلا ۴۵+ درجه) و یک بار دیگر با چرخش به چپ (مثلا ۴۵- درجه) ربات را از کنار مانع عبور دهد. این ناسازگاری باعث ایجاد مشکل در آموزش می‌شود. الگوریتم یادگیری باناظر برای این وضعیت یک عمل را باید تعیین نماید. واضح است که هر کدام را انتخاب کند خطا برای داده دیگر زیاد می‌شود و از آنجا که روش بر اساس کمینه کردن مجموع مربعات خطا می‌باشد، خروجی تعیین شده برای این وضعیت عددی نزدیک به صفر است. این خروجی به معنی حرکت مستقیم ربات به طرف جلو در این وضعیت و در نتیجه برخورد با مانع است. شکل ۱ بیانگر مورد مذکور می باشد. در دو قسمت "الف" و" ب" ربات توسط ناظر یکبار از سمت راست و یکبار از سمت چپ عبور داده شده است. نتیجه یادگیری با این داده ناسازگار حرکت مستقیم به سمت هدف و برخورد به مانع می باشد که در قسمت "ج" از شکل ۱ نشان داده شده است.



الف : چرخش به راست ب: چرخش به چپ



ج: حرکت مستقیم به طرف جلو و برخورد با مانع

شکل۱: تاثیر سوء مشکل ناسازگاری در داده‌های آموزشی.

[1]-Global algorithms

با توجه به ضعف‌های اشاره شده در یادگیری باناظر، استفاده از روش‌های هوشمند برای یادگیری ربات‌ها گسترش یافت. یادگیری تقویتی[1] یک الگوریتم مدرن هوشمند است که به جهت دارا بودن قابلیت‌هایی همچون عدم نیاز به خروجی مطلوب، آموزش تنها با استفاده از یک معیار اسکالر راندمان، امکان آموزش برخط، و درجه کاوش بالا گزینه مناسبی جهت تنظیم پارامترهای کنترلگر ربات می‌باشد. در واقع در یادگیری تقویتی به عامل گفته نمی‌شود که عمل صحیح در هر وضعیت چیست، و فقط با استفاده از یک معیار اسکالر که سیگنال تقویتی نامیده می شود خوب یا بد بودن عمل به عامل نشان داده می شود. عامل موظف است با در دست داشتن این اطلاعات، یاد بگیرد که بهترین عمل کدام است. این ویژگی یکی از نقطه قوتهای خاص الگوریتم یادگیری تقویتی است[۸]. اما از جنبه دیگر، دو چالش پیش روی یادگیری تقویتی زمانبر بودن و کند بودن آموزش در آن است. این مشکل در مسائل ناوبری ربات هم که معمولا فضای حالت بزرگ است بطور جدی مشهود است. مسأله دیگر این است که امکان تنظیم همه پارامترهای کنترلگر (پارامترهای توابع عضویت مقدم در کنترلگرهای فازی یا پارامترهای وزن در لایه‌های ابتدایی کنترلگرهای عصبی) در الگوریتم‌های یادگیری تقویتی پیوسته که از آنها برای کنترلگر ربات استفاده شده است، وجود ندارد.

یک ایده سودمند برای بهره‌گیری از مزایا و کاهش ضعف‌های دو روش یادگیری تقویتی و یادگیری باناظر، استفاده از ترکیب این دو روش یادگیری می باشد. در [۱۱] از یادگیری باناظر برای تخمین اولیه احتمال انتخاب عمل استفاده شده است. نویسنده به دنبال روش ترکیبی از یادگیری تقویتی و یادگیری باناظر خطی است که از یادگیری باناظر خطی برای تولید سیاست انتخاب عمل در یادگیری تقویتی استفاده شده است. لذا انتخاب عمل در روش یادگیری کیو[2] با توجه به احتمال انتخاب عمل‌هایی که از داده آموزشی بدست آمده است انجام می شود. روش مذکور در مسأله سیستم مکالمه بکار گرفته شده است. در [۱۲] یادگیری تقویتی باناظر برای مسئله دنبال کردن خط در ربات متحرک استفاده شده است و از دانش ناظر بعنوان دانشی که می‌تواند برای تصمیم در خصوص کاوش در مرحله انتخاب عمل استفاده شود، بهره برده شده است. در [۱۳] ناظر با استفاده از کنترلگر PID عملی را برای هر حالت انتخاب می‌نماید. سپس در هنگام انتخاب عمل، در روش یادگیری تقویتی عمل انتخاب شده توسط کنترلگر PID شانس بالاتری برای انتخاب خواهد داشت. تابع ارزش عمل هم بر اساس روش یادگیری کیو به روز رسانی می شود. در [۱۴] نیز از ترکیب یادگیری با ناظر و یادگیری تقویتی برای مساله حرکت ربات انسان نما به سمت شارژر و اتصال به آن استفاده شده است. در اینجا سعی شده ارزش اولیه عملها از یادگیری تقویتی استفاده شود. البته فضای عملها گسسته است و کلا چهار عمل برای ربات در نظر گرفته شده است.

ایده های ارائه شده در مراجع فوق برای فضای حالت گسسته و یا فضای عمل گسسته هستند، در حالیکه توجه ما در این مقاله بر روی فضای حالت – عمل پیوسته است.

در [۱] ایده ای برای فضای حالت و عمل پیوسته آمده است؛ در آن از داده آموزشی تولید شده توسط ناظر برای تنظیم اولیه پارامترهای بخش عملگر در معماری عملگر- نقاد[3] [۱۵] استفاده شده است. در این مرجع ابتدا توسط یادگیری باناظر مقدار عمل برای هر حالت پیشنهاد می‌شود و سپس با استفاده از یادگیری تقویتی، مقدار نهایی پیرامون مقدار پیشنهادی تنظیم می شود. روش مذکور دو ضعف عمده دارد:

۱- اثر سوء ناسازگاری داده اشاره شده در بالا، باعث خطای زیاد در خروجی تنظیم شده توسط یادگیری باناظر می گردد.

۲- ضعف عدم کاوش مناسب در معماری عملگر-نقاد [۱۵]، علاوه بر ضعف ذاتی معماری عملگر-نقاد در این خصوص از آنجا که مرحله تنظیم با یادگیری تقویتی تنظیمات پیرامون مقدار تنظیم شده با روش یادگیری با ناظر صورت می‌گیرد، این ضعف تشدید شده است.

در کار قبلی، ما در مقاله [۱۶] ایده ای شبیه به روش فوق را در معماری نقاد-تنها بکار گرفتیم. بدین صورت که از داده های آموزشی با ناظر برای تنظیم اولیه توابع عضویت ورودی سیستم فازی استفاده شده است و آنگاه پارامترهای تالی کنترلگر فازی بصورت برخط با استفاده از روش یادگیری سارسای فازی[4] (FSL) [۹] که یک روش یادگیری تقویتی فازی[5] (FRL) با معماری نقاد-تنها[6] است تنظیم شده است. هرچند مشکل عدم کاوش بخاطر استفاده از معماری نقاد-تنها مرتفع شده است لیکن هیچ آموزشی در خصوص عمل خروجی مناسب برای هر وضعیت صورت نگرفته است و در واقع مقادیر تالی کنترلگر فازی تنها با روش FSL تنظیم می‌گردند. بهمین دلیل بهبود بدست آمده فاحش نیست.

در اینجا روشی جدید برای ترکیب یادگیری باناظر و یادگیری تقویتی فازی با معماری نقاد-تنها ارائه می شود. لازم به ذکر است دو معماری معروف استفاده شده در FRL، معماری نقاد-تنها و عملگر –نقاد می باشند. از مزایای معماری نقاد-تنها پتانسیل بالا در برقراری تعادل بین کاوش و بهره برداری از تجربیات است. لذا این معماری برای مسائلی که نیاز به کاوش بالا دارند مانند ناوبری ربات مناسب می‌باشد. دو الگوریتم یادگیری سارسای فازی [۹] و یادگیری کیو فازی[7] (FQL) [۴] بر اساس معماری نقاد-تنها ارائه شده‌اند. برای روش FQL نه تنها هیچ قضیه یا لمی در جهت همگرایی آن وجود ندارد بلکه مثالهای واگرایی [۹] آن نیز موجود می باشد. لیکن در [۹] قضایای مربوط به همگرایی و اثبات نقاط ایستای روش FSL آمده است لذا الگوریتم یادگیری تقویتی پیوسته استفاده شده در این مقاله روش FSL می باشد.

[1]-Reinforcement Learning
[2]-Q-learning
[3]-Actor-Critic
[4]-Fuzzy Sarsa Learning
[5]-Fuzzy Reinforcement Learning
[6]-Critic-only
[7]-Fuzzy Q- Learning

ایده آن است که با کمک دانش ناظر بجای تعیین یک عمل برای هر حالت، ارزش اولیه برای اعمال ممکن کنترلگر تعیین می‌شود. سپس با کمک یادگیری تقویتی، بصورت برخط تنظیم نهایی درجهت بهبود کارایی صورت می‌گیرد. این ترکیب باعث ایجاد تسریع در فرایند یادگیری، بهبود کیفیت آموزش، و کاهش تعداد برخوردهای ربات به موانع و همگرایی سریعتر در حین آموزش می‌شود. براساس بررسی‌های ما، این کار اولین روش ارائه شده برای ترکیب یادگیری باناظر و یادگیری تقویتی فازی پیوسته با معماری نقاد- تنها می‌باشد.

ساختار مقاله به شرح زیر است. دربخش دوم الگوریتم FSL شرح داده می شود. بخش سوم نحوه طراحی ساختار کنترلگر فازی را شرح می‌دهد. در بخش چهارم ایده مقاله برای ترکیب یادگیری تقویتی و یادگیری باناظر را شرح می‌دهیم. بخش پنجم به پیاده‌سازی و شبیه سازی کار پرداخته است. در بخش آخر بحث و نتیجه‌گیری آمده است.

۲- یادگیری سارسای فازی(FSL)

از آنجا که FSL از لحاظ تحلیل ریاضی و عملکرد نسبت به FQL ارجحیت دارد، بعنوان الگوریتم پایه در فرآیند یادگیری در کار ما انتخاب شده است. این الگوریتم برخلاف FQL که مستقل از سیاست می باشد، یک روش وابسته به سیاست است که تالی قاعده سیستم فازی را بصورت برخط تنظیم می نماید.

الگوریتم FSL از ترکیب سیستم‌های فازی بعنوان یک تقریب زننده تابعی خطی با روش سارسا [۱۰] حاصل شده است. یک سیستم فازی سوگنو مرتبه صفر با n ورودی و یک خروجی و R قاعده به فرم زیر را در نظر بگیرید:

$$\begin{aligned} \text{Ri}: \text{If } x_1 \text{ is } L_{i1} \text{ and} \cdots \text{and } x_n \text{ is } L_{in} \text{ then} \\ o_{i1} \text{ with value } w^{i1} \\ \text{or } o_{i2} \text{ with value } w^{i2} \\ \vdots \qquad\qquad \vdots \\ \text{or } o_{im} \text{ with value } w^{im} \end{aligned}$$

که در آن $s = x_1 \times \cdots \times x_n$ بردار n بعدی متغیرهای حالت ورودی، $L_i = L_{i1} \times \cdots \times L_{in}$ شامل n مجموعه فازی محدب نرمال با مرکزهای یکتا برای i امین قاعده، m تعداد عمل‌های گسسته ممکن برای هر قاعده، o_{ij} j امین عمل کاندید در قاعده i ام و w^{ij} مقدار ارزش تقریب زده شده برای عمل j ام در قاعده i ام است. در هر قدم زمانی برای هر قاعده، یک عمل از میان m عمل کاندید شده برای تالی قاعده بر مبنای مقدار وزن آن عمل انتخاب می‌شود و آنگاه عمل نهایی از ترکیب وزن دار این عمل‌ها حاصل می‌گردد. هدف آموزش به روز رسانی بر خط مقادیر وزن w^{ij} با توجه به سیگنال تقویت دریافت شده است، بگونه‌ای است که بهترین انتخاب عمل بر مبنای آن‌ها حاصل گردد [۹].

شدت آتش هر قاعده از حاصلضرب درجه‌های تطابق مقدم قاعده برای ورودی‌های مختلف بدست می‌آید و خروجی سیستم a و مقدار ارزش- عمل تقریب زده شده برای عمل a در حالت s که با $\tilde{Q}(s,a)$ نشان داده می‌شود، بصورت ذیل محاسبه می‌شوند[۹]:

$$a_t(s_t) = \sum_{i=1}^{R} \mu_i(s_t) o_{ii^+} \qquad (۱)$$

$$\tilde{Q}_t(s_t, a_t) = \sum_{i=1}^{R} \mu_i(s_t) w_t^{ii^+} \qquad (۲)$$

پس از محاسبه عمل نهایی a_t و اِعمال آن، محیط به حالت جدید s_{t+1} رفته و عمل جدید a_{t+1} با توجه به مقادیر وزن فعلی w_t انتخاب می‌شود. ضمناً سیگنال تقویتی r_{t+1} از محیط دریافت می‌گردد. آنگاه مقادیر پارامترهای وزن هر قاعده بصورت زیر به روز رسانی می‌شوند [۹]:

$$\Delta w_{t+1}^{ij} = \begin{cases} \alpha_t \times \Delta\tilde{Q}_t(s_t, a_t) \times \mu_i(s_t) & if \;\; j = i^+ \\ 0 & \text{otherwise} \end{cases} \qquad (۳)$$

که α نرخ آموزش و γ فاکتور نزول و $\Delta\tilde{Q}$ خطای تقاضل موقتی ارزش- عمل است و بصورت ذیل محاسبه می‌گردد:

$$\Delta\hat{Q}_t(s_t, a_t) = r_{t+1} + \gamma \hat{Q}_t(s_{t+1}, a_{t+1}) - \hat{Q}_t(s_t, a_t) \qquad (۴)$$

قدم‌های الگوریتم FSL بصورت خلاصه در زیر آمده است[۹]:

۱- مشاهده حالت S_{t+1} و دریافت سیگنال تقویتی r_{t+1}.
۲- انتخاب یک عمل مناسب در هر قاعده با استفاده از روش انتخاب عمل بیشینه نرم.
۳- محاسبه عمل نهایی a_{t+1} و مقدار تقریبی تابع ارزش- عمل $\hat{Q}_t(s_{t+1}, a_{t+1})$ با استفاده از (۱) و(۲).
۴- محاسبه $\Delta\hat{Q}$ و بروزرسانی w با استفاده از (۳) و(۴).
۵- محاسبه مقدار تقریبی جدید $\hat{Q}_{t+1}(s_{t+1}, a_{t+1})$ با استفاده از(۴)
۶- اجرای عمل نهایی .
۷- $t \leftarrow t+1$ و بازگشت به مرحله اول.

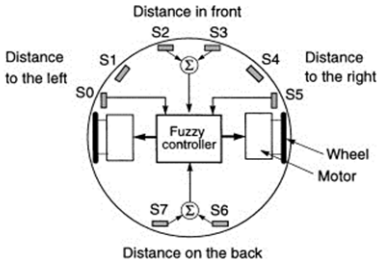
۳- طراحی کنترلگر برای ناوبری ربات

برای بررسی ایده مورد بحث از ربات خپرا[1] که یک ربات مینیاتوری برای فعالیت‌های آزمایشگاهی و تحقیقاتی ساخته شده توسط شرکت سوییسی K-Team [۱۷] است، استفاده می‌کنیم. قابلیت فراوان این ربات و اندازه مناسب آن جهت فعالیت‌های آزمایشگاهی، منجر به استقبال گسترده محققین در استفاده از این ربات در ارزیابی روش‌های خود شده است[۱۸].

پیرامون ربات خپرا هشت حسگر مادون قرمز که هر یک دارای یک فرستنده و گیرنده هستند، وجود دارد. هر دو حسگر در یک وجه ربات نصب شده است (شکل۲). محدوده عملکرد مؤثر حسگرهای این ربات

[1]-Khepera robot

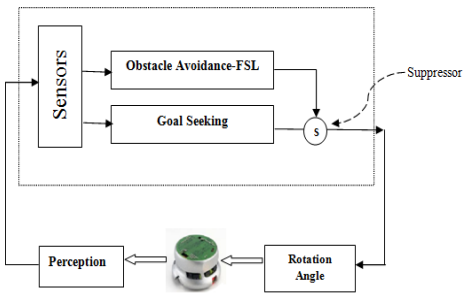
بین۱ تا ۵ سانتیمتر است و مقدار خروجی آنها با فاصله ربات تا مانع رابطه عکس دارد. هر چه ربات به مانع نزدیک‌تر گردد، مقدار خروجی حسگر بیشتر و هر چه دورتر گردد، مقدار کمتری خواهد داشت.



شکل۲: موقعیت سنسورها در ربات مینیاتوری خپرا.

بر روی هر چرخ ربات یک رمزگذار[1] نصب شده است که با شمارش پالس‌های حاصل از رمزگذارها می‌توان مسافت طی شده توسط هر چرخ را محاسبه نمود. همچنین بسته به کاربرد می‌توان تجهیزات جانبی دیگری همچون دوربین، چنگک و غیره بر روی ربات نصب نمود [۱۷].

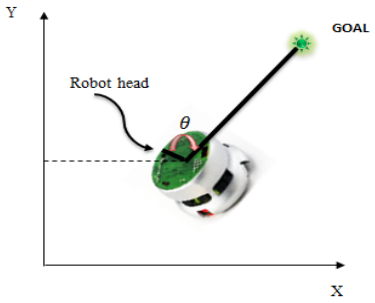
در اینجا هدف آن است که اگر ربات در مجاورت موانع قرار دارد، بدون برخورد به مانع با توجه به موقعیت هدف از موانع عبور کند، و زمانی که پیرامون ربات مانعی وجود ندارد، ربات بطور حریصانه، به سمت هدف چرخیده به سمت آن حرکت کند. برای عملکرد بهتر ربات هنگام حرکت و نیزکاهش پیچیدگی سیستم، از معماری رده‌بندی۲ که بروکس در [۱۹] ارائه نموده است، استفاده می‌نماییم. با استفاده از این معماری برای ربات دو رفتار در نظر گرفته می‌شود. یک رفتار بنام "اجتناب از موانع" برای زمان‌هایی که ربات نزدیک موانع است و رفتار دیگر "پیگیری هدف" برای زمانی که پیرامون ربات مانعی وجود ندارد.



شکل۳: معماری مبتنی بر رفتار.

شکل۳ طراحی انجام شده در این مقاله بر مبنای معماری مذکور را نشان میدهد. خروجی رفتار "پیگیری هدف" همان‌طور که در شکل ۴ آمده است به سادگی با محاسبه زاویه پیشانی ربات با هدف بدست می آید. بدین معنی که ابتدا مقدار اختلاف زاویه بدست آمده و سپس ربات به اندازه زوایه مذکور چرخیده و پس از همراستا شدن بطور مستقیم به سمت هدف می رود. همان طور که در شکل ۳ نیز مشخص می باشد، زمانی که ربات نزدیک مانع می‌شود رفتار "پیگیری هدف" غیرفعال می-شود و خروجی حاصل از ماژول "اجتناب از موانع" به ربات اعمال می-شود. وظیفه این ماژول تعیین زاویه حرکت ربات در هر قدم زمانی به گونه ای است که ضمن پرهیز از برخورد به موانع در جهت نزدیک شدن به هدف، ربات حرکت کند. توجه شود از آنجا که در این ماژول خروجی تولید شده با در نظر گرفتن دو مورد اجتناب از موانع و نزدیک شدن به هدف تولید می شود. لذا دیگر مانند دیگر کارهای مرتبط در این زمینه[۲۰] نیازی به یک ماژول برای ترکیب خروجی های رفتارها نیست. از این رو هزینه محاسبات و پیچیدگی سیستم کاهش یافته است.

پیشنهاد ما برای طراحی ساختار این کنترلگر، یک کنترلگر فازی سوگنو مرتبه صفر می باشد. ساختار این کنترلگر بصورتی در نظر گرفته می شود که با ساختار استفاده شده در FSL همخوانی داشته باشد.



شکل ۴: محاسبه زاویه چرخش ربات برای رفتار پیگیری هدف.

کنترلگرمورد نظر دارای چهار ورودی (سه ورودی اول بعنوان معیار فاصله ربات با مانع در یکی از سه جهت راست، جلو، و عقب هستند، و ورودی چهارم زاویه پیشانی ربات با هدف) و یک خروجی (مقدار زاویه چرخشی پیشانی ربات در هنگام نزدیکی به موانع) می‌باشد. توابع عضویت ورودی این کنترلگر بشکل گوسی در نظر گرفته شده اند. در هر بعد ورودی بترتیب ۲،۳،۲و۲ مجموعه فازی تعریف شده است. با توجه به تقسیم بندی انجام شده در هر بعد ورودی، کنترلگر دارای ۲۴ قاعده می باشد. مقدار خروجی هر قاعده یک مقدار ثابت است که باید از مجموعه عملهای کاندید در نظر گرفته شده برای هر قاعده($A=\{O_{i1},O_{i2},...,O_{im}\}$) انتخاب شود. عمل مناسب برای هر قاعده از این مجموعه عمل کاندید تعیین می‌گردد. لذا هدف از آموزش تعیین عمل مناسب از میان مجموعه عمل کاندید برای تالی هر قاعده است.

۴- یادگیری سارسای فازیِ باناظر

در این بخش روش جدیدی برای تعیین عمل مناسب از میان مجموعه عمل‌های کاندید ممکن برای تالی هر قاعده، در ساختار کنترلگر فازی سوگنو مرتبه صفر ارائه می‌گردد. روش ارائه شده که ترکیبی از یادگیری باناظر و یادگیری تقویتی است شامل دو مرحله می‌باشد:

[1]-Encoder
[2]-Subsumption

در مرحله اول ابتدا با حرکت ربات در محیط توسط ناظر، داده آموزشی جمع‌آوری می‌شود. در اینجا بر خلاف روش‌های موجود که از داده آموزشی برای تعیین عمل مشخص برای هر حالت استفاده می‌شود، یک روش جدید جهت استفاده از داده آموزش برای ارزش گذاری عمل‌های ممکن در هر حالت ارائه می‌شود. بدین صورت که این داده آموزشی برای مقداردهی اولیه ارزش هر عملِ کاندید w^{ij} (معرفی شده در بخش دوم) در تالی هر قاعده کنترلگر فازی استفاده می‌شود. از این رو هدفی که ما بدنبال آن هستیم تعیین ارزش برای هر خروجی انتخاب شده در هر حالت توسط ناظر است.

بدین مفهوم که مثلا اگر ناظر عملهای متفاوتی را در یک وضعیت خاص در دفعات مجزا انتخاب کند، متناسب با تعداد انتخاب هر عمل در آن وضعیت خاص، به آن عمل ارزش داده شود. از آنجا که خروجی نهایی سیستم فازی از ترکیب وزن دار تالی انتخاب شده در هر قاعده بدست می آید. لازم است برای هر خروجی یک ترکیب ممکن از عمل-های کاندید هر قاعده به گونه‌ای که ترکیب این اعمال بتواند منجر به مقداری نزدیک به آن خروجی شود، پیدا نموده و آنگاه ارزش آن عمل-ها افزایش یابد.

هر نمونه p ام از داده‌های جمع‌آوری شده شامل جفت داده ورودی- خروجی (x_p , y_p) را در نظر بگیرید که x_p ورودی کنترلگر و y_p خروجی پیشنهاد شده توسط ناظر می‌باشد. قدمهای زیر برای تعیین ارزش اولیه عمل‌های کاندید در هر قاعده (w^{ij})، برای هر نمونه p ام از داده‌ها (x_p , y_p) دنبال می‌شود.

۱- برای ورودی x_p، چهار قاعده غالب (قاعده‌هایی که بیشترین میزان شدت آتش(μ) را دارند) را انتخاب می‌نماییم. این قواعد با سمبلهای l_1 و l_2 و l_3 و l_4 نشان داده می‌شوند، طوری‌که:
$\mu_{l4} < \mu_{l3} < \mu_{l2} < \mu_{l1}$.

۲- y_p بر μ_{l1} (بیشترین شدت آتش) تقسیم می شود.

۳- نتیجه تقسیم با هر یک از عمل‌های کاندید (O_{l1j}) مقایسه می-شود. سپس نزدیک ترین عمل به نتیجه تقسیم انتخاب شده و k_1 بعنوان اندیس آن عمل در نظر گرفته می‌شود.

۴- شمارنده c_{ij} را برای نشان دادن دفعات انتخاب j امین عمل کاندید، درi امین قاعده بکار می‌بریم. در این مرحله مقدار شمارنده عمل k_1 ام در قاعده l_1، (c_{l1k1}) یک واحد افزایش می یابد.

۵- عمل k_1 ام از مجموعه عملهای کاندید قاعده l_1 در شدت آتش این قاعده (μ_{l1}) ضرب شده و این حاصلضرب را از y_p کم می کنیم.

$$(y'_p = y_p - \mu_{l1} \times a_{l1k1}) \qquad (۵)$$

۶- مقدار y'_p را با مقدار y_p اولیه و مقدار شدت آتش قاعده l_2 (μ_{l2}) را با مقدار شدت آتش قاعده l_1 (μ_{l1}) جایگزین می کنیم.

۷- مراحل بالا را بطور کامل برای μ_{l2} و سپس برای μ_{l3} (مانند قبل با جایگزینی مقدار μ_{l3} بجای μ_{l2}) و μ_{l4} (با جایگزینی مقدار μ_{l4} بجای μ_{l3}) تکرار می کنیم.

۸- در نهایت وقتی که مراحل بالا برای همه نمونه داده‌های جمع آوری شده انجام شد، ارزش عمل j امین از قاعده i امین بصورت زیر مقداردهی اولیه می کنیم.

$$w^{ij} = \frac{c_{ij}}{(\sum_j c_{ij})^2} \qquad (۶)$$

شبه کد روش پیشنهاد شده در شکل ۵ آمده است.

Initialize counter (C_{ij}) for j-th candidate action in i-th rule with zero, assign set A_{set}
Foreach (x_p,y_p),x_p is input of the controller and y_p is suggested output by supervisor **do**
  Select four rules that have highest firing strength (l_1, l_2, l_3, l_4)
  $output = y_p$, $i=1$
  repeat for each rule L_i
    Calculate firing strength (μ_{li})
    $a = [\frac{output}{\mu_{li}}]$
    closest action to the division result is selected and indexed by k_l
    $c_{ij} = c_{ij} + 1$
    $y'_p = output - \mu_{li} \times a_{ilk1}$
    $output = y'_p$
  until $i=4$
end
foreach rule i
  foreach candidate action j
    $w^{ij} = \frac{c_{ij}}{(\sum_j C_{ij})^2}$
  end
end

شکل ۵: شبه کد روش پیشنهاد شده برای یافتن ارزش اولیه عمل‌های کاندید.

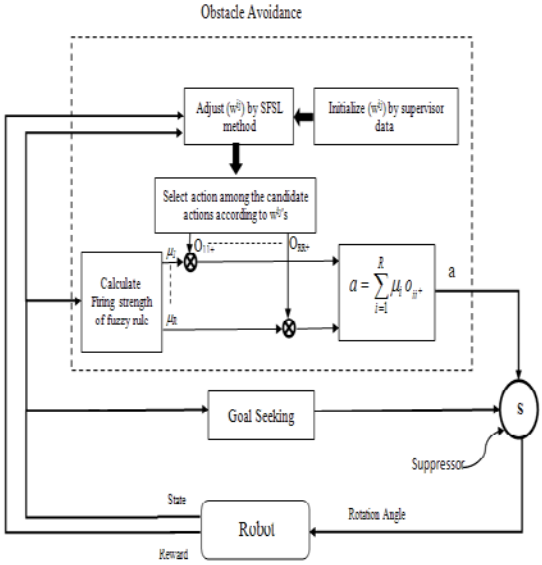
پس از تعیین مقدار اولیه w^{ij} ها در مرحله دوم از الگوریتم FSL برای تنظیم برخط تالی قواعد کنترلگر فازی که مقدار ارزش عمل‌های آن(w^{ij}) بصورت بالا مقداردهی اولیه شده است، استفاده می‌کنیم. روش ترکیبی مذکور را یادگیری سارسای فازیِ باناظر [1] (SFSL) می-نامیم. بلوک دیاگرام SFSL در شکل۶ آمده است. بطور خلاصه روش SFSL شامل مراحل زیر می شود:

۱- حرکت ربات در محیط و جمع آوری داده های آموزشی.

۲- مقداردهی اولیه ارزش عمل‌های کاندید با روش ارائه شده (شکل۵).

۳- تنظیم نهایی مقدار تالی قواعد با استفاده از FSL.

[1]–Supervised Fuzzy Sarsa Learning



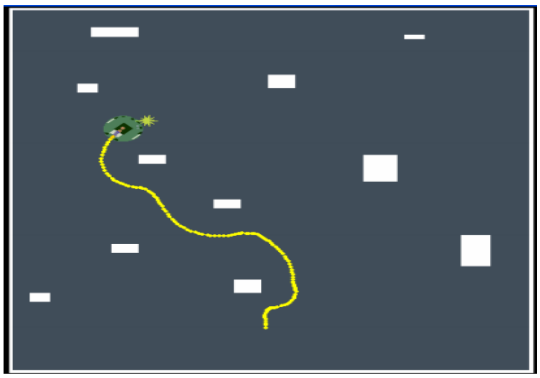
شکل ۶: نمودار بلوکی روش SFSL .

۵- شبیه سازی

شبیه‌سازهای متعددی برای ربات خپرا موجود است. در این میان، شبیه‌ساز KiKS که یک شبیه‌ساز ربات خپرا درمحیط برنامه نویسی MATLAB است، برای مسأله ناوبری ربات استفاده می‌شود که این شبیه-ساز مورد توجه بسیاری از محققین قرار گرفته است [۲۱]. در این پژوهش نیز از این شبیه‌ساز استفاده شده است. در ابتدای کار لازم است محیط‌های لازم برای شبیه سازی آماده گردد. برای هر محیط ابعاد آن، موقعیت و شکل موانع، موقعیت شروع حرکت ربات، و موقعیت هدف تعیین می-گردد. برای این کار از واسط گرافیکی شبیه‌ساز و دستورات مرتبط با آن در شبیه‌ساز استفاده شده است.

ابعاد محیط آموزشی ۸۲۰×۸۲۰ میلی متر مربع است که موانع مختلفی با شکل‌های متفاوت در آن قرار گرفته‌اند. شکل ۷ محیط آموزش ربات را با موقعیت‌های مختلف شروع حرکت ربات و هدف نشان می‌دهد. در این محیط، مستطیل‌های سفید رنگ موانع و ستاره زرد رنگ، هدف می‌باشد.

هر رویداد[1] در بخش آموزش شامل شروع از مبدأ و حرکت ربات تا رسیدن به هدف است. موقعیت هدف و شروع حرکت ربات در هر رویداد متفاوت می باشد. بخش آموزش در صورتیکه ربات به کران بالای تعداد حرکت‌ها که ۵۰۰ است و یا اینکه ۱۰ بار بطورمتوالی بدون شکست به هدف برسد، به پایان می‌رسد. شماره رویدادها در پایان بخش آموزش بعنوان معیار زمان آموزش[2] LDI در نظر گرفته می شود.



شکل ۷: نمونه ای از محیط آموزش در شبیه ساز KiKS.

با توجه به مطالب ذکر شده در بخش سوم، معماری رده‌بندی برای حرکت ربات در این مقاله پیشنهاد شد و هدف اصلیِ آموزش، تنظیم پارامترهای کنترلگر فازی برای ماژول"اجتناب از موانع " در معماری طراحی شده شکل۳، می‌باشد. سه ورودیِ اول تعریف شده در بخش سوم برای کنترلگر ربات با ترکیب خروجی های حسگرهای مادون قرمز هر کدام از وجوه راست، جلو و چپ بصورت زیر حاصل می‌شوند:

$$d_{face}(t) = 1 - \frac{\max(s_{face,1}, s_{face,2})}{1024} \qquad face \in \{Left, Front, Right\} \qquad (۷)$$

که $s_{face,1}$ و $s_{face,2}$ مقدار خروجی حسگرهای یک و دو برای وجه مشخص شده در اندیس آن (چپ، جلو، و یا راست) می‌باشند. لازم به ذکر است خروجی حسگرها که در شبیه ساز عددی بین ۰ تا ۱۰۲۳ است بصورت پیش فرض همراه با مقداری نویز جمع شده است. مقدار صفر برای وقتی است که حسگر هیچ مانعی را در محدوده خود حس نکند و مقدار ۱۰۲۳ برای هنگامی است که حسگر تقریبا به مانع چسبیده است. ورودی چهارم زاویه پیشانی ربات با هدف است، که آن را با $\theta(t)$ نشان می‌دهیم و مقدار آن عددی بین ۱۸۰ و ۱۸۰- درجه می‌باشد.

چهار ورودی مذکور نرمالیز شده و سپس به کنترلگر وارد می گردند. خروجی کنترلگر زاویه‌ی چرخش پیشانی ربات است که عددی بین ۴۵- درجه تا ۴۵+ درجه در نظرگرفته شده است.

مرحله اول شامل حرکت ربات توسط ناظر در محیط آموزش شکل۷ با کمک جوی استیک[3] می باشد. ۱۲۰۰ جفت داده ورودی - خروجی در این مرحله جمع آوری شد. از این داده ها با استفاده از روش ارائه شده در بخش چهارم، مقدار اولیه ارزش عمل‌های کاندید (w^{ij}) در هر قاعده کنترلگر فازی مشخص گردید.

[1]-Episode
[2]-Learning Duration Index
[3]-Joystick

در مرحله دوم آموزش از الگوریتم FSL، برای تنظیم برخط پارامترهای کنترلگر استفاده می‌شود. در الگوریتم FSL معیار فاصله ربات با موانع که d نامیده می‌شود، بصورت زیر تعریف گردید:

$$d = \min(d_{face}) \quad face \in \{Left, Front, Right\} \qquad (۸)$$

هرگاه d صفر شود یک شکست[1] به حساب می‌آید. هرگاه فاصله مرکز ربات تا هدف به ۵۰ میلی متر برسد، به معنی رسیدن ربات به هدف است. سیگنال تقویتی را با توجه به نزدیکی به موانع و زاویه سر ربات با هدف بصورت زیر تعریف می‌نماییم:

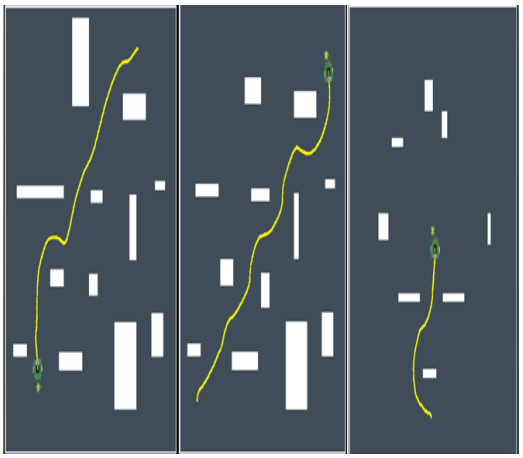
$$r(t) = \begin{cases} -1 & failure \\ -0.5 & d < 0.075 \\ \Delta/150 & \Delta > 0 \ \& \ d \geq 0.075 \\ -0.01 + \Delta/150 & \Delta \leq 0 \ \& \ d \geq 0.075 \\ 1 & goal \end{cases}$$
$$, \quad \Delta = |\theta(t-1)| - |\theta(t)| \qquad (۹)$$

برای تالی هر قاعده ۱۳ عمل کاندید بصورت زیردر نظر گرفته شده-است.

A= {-۴۵ ،-۳۰ ،-۲۰ ،-۱۵ ،-۱۰ ،-۵ ،۵ ،۱۰ ،۱۵ ،۲۰ ،۳۰ ،۴۵}

برای آموزش در این مرحله، پانصد جفت موقعیت تصادفی برای نقطه شروع حرکت ربات و هدف در محیط آموزشی شکل۷ تولید شدند. ده اجرای مستقل انجام پذیرفت. هر اجرا از دو بخش آموزش و تست تشکیل می گردد.

پس از پایان آموزش، برای ارزیابی عملکرد ربات، بخش تست انجام می پذیرد. در این بخش ۱۰ محیط جدید ایجاد شده که در ۸ محیط اول تنها موقعیت شروع حرکت ربات وهدف متفاوت است. شکل۸ این محیط‌ها را نشان می‌دهد. کیفیت عملکرد در محیط تست با معیارهای تعداد برخورد به موانع، و مسافت طی شده تا رسیدن به هدف ارزیابی می شود.



شکل۸: محیط‌های تست مختلف برای رویدادهای ۱ تا ۱۰ در بخش تست

جهت مقایسه روش ارائه شده با یک روش ترکیبی مشابه، الگوریتم CSLAFSL [۱۶] انتخاب شد. برای روش CSLAFSL از داده آموزشی بدست آمده توسط حرکت ربات توسط ناظر برای تنظیم درجه عضویت توابع عضویت ورودی قواعد فازی طبق روش بیان شده در مقاله [۱] استفاده شد و آنگاه در معماری رده بندی ارائه شده (شرح داده شده در بخش سوم) بکار گرفته شد. همچنین دو روش FSL [۹]، و FQL [۴] نیز در معماری رده بندی ارائه شده بکار رفتند و نتایج شبیه سازی آورده شده است. توجه شود که در این دو روش ارزش اولیهِ عمل‌های تالی ها (w^{ij}) صفر می باشند (مقدار دهی اولیه نشده‌اند).

نتایج شبیه سازی در جدول ۱ آورده شده است. ستون اول این جدول چهار الگوریتم یادگیری را نشان می‌دهد. ستون دوم این جدول متوسط LDI ها را در بخش آموزش نشان می‌دهد که مقدار آن از متوسط‌گیری بر روی ۱۰ اجرای مستقل بدست آمده است. ستون سوم و چهارم نشان دهنده متوسط تعداد برخوردهای ربات با موانع، بترتیب در بخش آموزش و تست می‌باشد. نهایتاً در ستون پنجم متوسط مسافت پیموده شده توسط ربات در بخش تست آورده شده است.

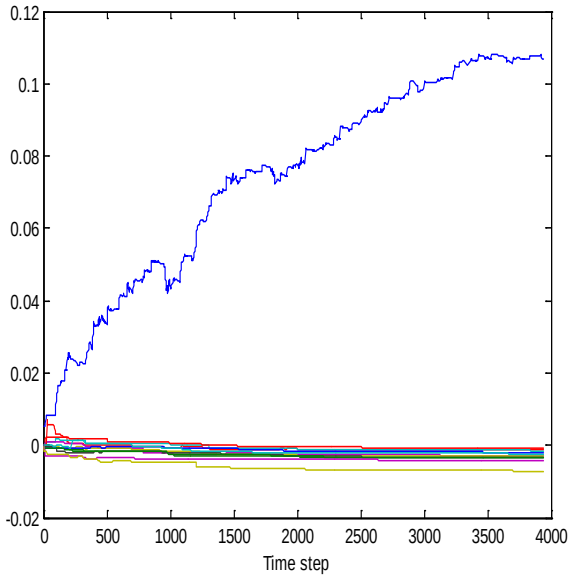
جدول ۱: نتایج شبیه سازی در مسأله ناوبری ربات.

Methods	Ave. LDI	Failure Rate 1	Failure Rate 2	Ave. Distance
SFSL	40	30.4	8.7	78.00
CSLAFSL	88	49.3	9.1	78.04
FSL	107	62.6	9.3	78.12
FQL	124	66.5	9.8	77.08

همانطور که از نتایج مشهود است عملکرد روش SFSL بطور قابل توجهی از سه روش دیگر بهتر است. این روش برای معیار Ave. LDI که نشانگر سرعت آموزش است، ۵۰ درصد بهتر از CSLAFSL، ۶۲ درصد سریعتر ازFSL و ۶۸ درصد سریعتر از FQL ،کنترلگر فازی را تنظیم می-کند. بعبارتی این روش سرعت زمان آموزش را حداقل ۶۰ درصد افزایش داده است. همچنین تعداد شکست ها در بخش آموزش در روش SFSL از سه روش بطور قابل توجهی (تقریبا ۵۰ درصد) کمتر شده است. تعداد شکست ها در بخش تست نیز در روش SFSL ، ۴ درصد کمتر از CSLAFSL ، ۶ درصد کمتراز FSL و ۱۱ درصدکمتر از FQL می‌باشد. از آنجا که هر چهار روش در همه تکرارها در مرحله تست به هدف رسیده اند مسافت طی شده تا هدف در آنها تقریباً یکسان است. جهت نمایش نحوه تغییرات مقادیر وزن عملهای کاندید، نمودار تغییرات مقدار ارزش (w^{ij}) عملهای کاندید در قاعده ۲۳ ام کنترلگر فازی در شکل ۹ آمده است. همانطور که دیده می‌شود ارزش مربوط به اولین عمل کاندید (مربوط به °۴۵-) بیشترین مقدار را دارد که پس از گذشت زمان کوتاهی از آموزش مقدار ارزش آن از بقیه عملها پیشی گرفته است. همچنین

[1] - Failure

ترتیب مقدار ارزش عملهای دیگر نیز پس از گذشت زمان کوتاهی از آموزش دیگر تغییر نمی‌کند.



شکل۹: نمودار تغییرات مقدار ارزش عملها در تالی قاعده ۲۳ ام کنترلگر

۶- بحث و نتیجه‌گیری

در این مقاله یک روش جدید برای ترکیب یادگیری باناظر و یادگیری تقویتی پیوسته پیشنهاد شد. جهت بررسی عملکرد روش مذکور از مسأله ناوبری ربات استفاده شد. در این روش فرمان‌های کنترلی هدایتِ ربات، براساس معماری رده‌بندی بروکس طراحی شد و فرمان نهایی از ترکیب دو رفتار "اجتناب از موانع " و "پیگیری هدف" بدست آمد. برای رفتار "اجتناب از موانع" یک کنترلگر فازیِ سوگنو مرتبه صفر طراحی شد. هدف یادگیری یافتن مقدار تالی مناسب برای هر قاعده این کنترلگر بود. روش ارائه شده از داده‌های آموزشی تولید شده که با کمک ناظر و از طریق حرکت دادن ربات در محیط بدست آمده بود، برای تقریب مقدار ارزش هر عمل کاندید استفاده نمود. در مرحله دوم آموزش، از الگوریتم FSL به عنوان الگوریتم یادگیری تقویتی پیوسته برای تنظیم نهایی مقدار تالی قواعد کنترلگر فازی بهره برده شد. نتایج شبیه سازی برای روش ارئه شده نشان داد که زمان آموزش و تعداد برخورد به موانع، کاهش قابل توجهی نسبت به سه روش CSLAFSL، FSL و FQL هنگامی که پارامترهای آن‌ها مقدار دهی اولیه نشده اند، دارد. لازم به توجه است که در اینجا بر خلاف روش‌های مرسوم که از یادگیری باناظر برای تعیین مقدار خروجی کنترلگر برای هر حالت استفاده می کنند، از داده‌های آموزشی بدست آمده، برای ارزش‌دهی به عمل‌های کاندید در هر حالت استفاده شد. بدین طریق نه تنها از اثر مخرب داده‌های ناسازگار جلوگیری بعمل آمد، بلکه از دانش موجود در این داده‌ها نیز سود برده شد. نکته قابل توجه دیگر استفاده از معماری رده‌بندی جهت ناوبری ربات بود. ساختار ارائه شده دیگر نیازی به یک ماژول برای ترکیب رفتارها ندارد و در هر لحظه تنها خروجی یک رفتار به ربات اعمال می شود، لذا هزینه محاسباتی و طراحی کاهش یافته است. در خصوص تحلیل ریاضی روش ارائه شده باید گفته شود که از آنجا که روش مذکور در واقع از داده باناظر برای مقدار دهی اولیه پارامترهای روش FSL استفاده کرده است. لذا تمام شرایط بیان شده در قضایای ارائه شده برای FSL در مقاله [۹] را دارد و تحلیل‌ها و قضایای ریاضی بیان شده در آن مرجع برای SFSL نیز برقرار است.

سپاسگزاری

این تحقیق با حمایت "صندوق حمایت از پژوهشگران کشور INSF" انجام شده است که بدینوسیله از آن مرکز محترم تشکر و قدردانی می گردد.

مراجع

[1] C. Ye, N. H. C. Yung, and D. Wang,“A fuzzy controller with supervised learning assisted reinforcement learning algorithm for obstacle avoidance,” IEEE Transaction Systems, Man, Cybernetics, vol. 33, no. 1, pp.17-27, Feb. 2003.

[2] T. Belker, M. Beetz, and A. Cremers, “Learning action models for the improved execution of navigation plans,” Robotics and Autonomous Systems, vol. 38, pp. 137-148, Mar. 2002.

[3] T. Fong, I. Nourbakhsh, and K. Dautenhahn,“A survey of socially interactive robots,” Robotics and Autonomous Systems, vol. 42, pp. 143-166, Mar. 2003.

[4] L. Jouffe, “Fuzzy inference system learning by reinforcement methods,” IEEE Trans. Syst., Man, Cybern. C, vol. 28, no.3, pp. 338-355, Aug. 1998.

[5] H. R. Beom, and H. S. Cho, “A sensor-based navigation for a mobile robot using fuzzy logic and reinforcement learning,” IEEE Trans. Syst., Man, Cybern., vol. 25, no. 3, pp. 464–477, Mar. 1995.

[6] K. Macek, I. Petrovic, and N. Peric,“A reinforcement learning approach to obstacle avoidance of mobile robots,” Proc. IEEE Int. Conf. Advanced Motion Control, vol.1, pp. 462-466, 2002.

[7] J. S. R. Jang, C. T. Sun, and E. Mizutani, "Neuro-Fuzzy and soft computing," Prentice-Hall, 1997.

[۸] م.کلامی هریس، ن.پریز، م. ب.نقیبی سیستانی " بررسی یادگیری تقویتی و خواص سیاست بهینه در مسائل جدولی با استفاده از روش‌های کنترل دیجیتال " ، مجله کنترل، جلد ۳، شماره ۱، بهار ۱۳۸۸

[9] V. Derhami, V. Majd, and M. Nili Ahamadabaadi “Fuzzy Sarsa learning and the proof of its

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

[۱۶] ف.فتحی نژاد، و.درهمی” ترکیب یادگیری با ناظر با یادگیری تقویتی برای ناوبری ربات“ ، هفدهمین کنفرانس ملی سالانه انجمن کامپیوتر ایران، صفحه: ۱۱۵-۱۱۹، اسفند ۱۳۹۰

[17] H. Maaref, and C. Barret “Sensor-based navigation of a mobile robot in an indoor environment” Robotics and Automation systems, vol. 38, pp. 1-18, Jan. 2002.

[18] E. O. Ari, I. Erkmen, and A. M. Erkmen, “A FACL controller architecture for a grasping snake robot,” Proc. IEEE Int. Conf. Intelligent Robots and Systems, pp. 1748-1753, 2005.

[19] R. A. Brooks, “A robust layered control system for a mobile robot, “Journal of Robotics and Automation, vol. 2, pp. 14-23, Mar.1986.

[20] K.Anam, Prihastono, H. Wicaksono, R.Effendi, S. Kuswadi5, “Hybridization of Fuzzy Q-learning and Behavior-Based Control for Autonomous Mobile Robot Navigation in Cluttered Environment” International Joint ICROS- Conf SICE, pp. 1023 - 1028, Aug. 2009.

[21] T. Nilsson, “KIKS: KIKS Is a Khepera Simulator” http:// www.kiks.f2s.com.

stationary points” Asian Journal of Control, vol. 10, No. 5, pp. 535-549, September 2008.

[11] J. Henderson, O.Lemon, K. Georgila, “Hybrid Reinforcement/Supervised Learning for Dialogue Policies from communicator data,” In IJCAI workshop on Knowledge and Reasoning in Practical Dialogue Systems, 2005.

[12] R. Iglesias, C. V. Regueiro, J. Correa, S. Barro, “supervised reinforcement learning: application to a wall following behaviour in a mobile robot,” Lecture Notes in Computer Science, vol. 1416, pp. 300-309, 1998.

[13] L.Lin, H.Xie, D.Zhang, L.Shen, “Supervised Neural Q_learning based Motion Control for Bionic Underwater Robots,” Journal of Bionic Engineering, vol.7, pp. 177-184, Sept.2010.

[14] N.Navarro-Guerrero, C.Weber, P.Schroeter, S.Wermter, “Real-world reinforcement learning for autonomous humanoid robot docking”, Robotics and Autonomous Systems, vol. 60, pp.1400-1407.Nov.2012.

[15] Su.Shun-Feng, H.Sheng-Hsiung, “Embedding Fuzzy Mechanisms and Knowledge in Box-Type Reinforcement Learning Controllers,” IEEE Transaction System, Man, Cybernetic. vol. 32, pp. 645-653, Oct. 2002.