

طراحی کنترل گر فازی با قابلیت تنظیم برخط برای کنترل بینامبناى بازوی ربات

فاطمه السادات آبادیان زاده^۱، ولی درهمی^۲، مهدی رضاییان^۳

^۱ دانشجوی کارشناسی ارشد، پردیس فنی و مهندسی، گروه مهندسی کامپیوتر، دانشگاه یزد، abadian.fatemeh@stu.yazd.ac.ir

^۲ دانشیار، پردیس فنی و مهندسی، گروه مهندسی کامپیوتر، دانشگاه یزد، vderhami@yazd.ac.ir

^۳ استادیار، پردیس فنی و مهندسی، گروه مهندسی کامپیوتر، دانشگاه یزد، mrezaeian@yazd.ac.ir

دریافت: 1395/11/30 ویرایش اول: 1395/10/13 ویرایش دوم: 1396/1/16 ویرایش سوم: 1396/4/15 پذیرش: 1396/1/18

چکیده: در کنترل بینامبنا ربات از اطلاعات استخراج شده از حسگر بصری برای کنترل حرکت ربات استفاده می‌شود. در روش‌های سنتی کنترل بینامبنا، مدل ربات و مدل دوربین مورد نیاز است. به دست آوردن این مدل‌ها زمان‌بر و گاهی اوقات غیرممکن است. بنابراین در تحقیقات اخیر از روش‌های هوشمند برای مقابله با این چالش استفاده می‌شود. در این پژوهش ابتدا از یک کنترل گر فازی ترکیبی برای کنترل بازوی ربات استفاده شده است. ورودی‌های بصری کنترل گر از طریق کینکت حاصل می‌شود و خروجی کنترل گر میزان چرخش زاویه موتورهای مفاصل ربات از موقعیت فعلی آن‌ها است. کنترل گر ترکیبی شامل دو کنترل گر می‌باشد. کنترل گر اول بر پایه مدل فازی معکوس است که تقریبی از مدل واقعی معکوس سیستم، با استفاده از داده‌های جمع‌آوری شده است. به منظور افزایش دقت، یک کنترل گر خبره فازی برای وقتی که موقعیت مجری نهایی نزدیک هدف است، طراحی شده است. از آنجا که تعیین دقیق پارامترهای کنترل گر خبره فازی ممکن نیست، همچنین به منظور تطبیق‌پذیری کنترل گر در برابر تغییرات جزئی در محیط کار، از معماری عملگر-نقاد که از جمله روش‌های یادگیری تقویتی است برای تنظیم پارامترهای آن استفاده شده است. کنترل گر ارائه شده بر روی بازوی ربات ARM_6AX18 پیاده‌سازی شده است. نتایج آزمایش‌های عملی نشان می‌دهد که با استفاده از روش ارائه شده، مجری نهایی به موقعیت هدف از پیش تعیین شده با دقت خوبی می‌رسد.

کلمات کلیدی: کنترل بینامبنا، سیستم فازی، مدل معکوس فازی، بازوی ربات، یادگیری تقویتی، عملگر-نقاد

Designing a Fuzzy Controller for Visual Servoing of a Robot Manipulator with Online Adjustment Capability

Fatemeh Abadianzadeh, Vali Derhami, Mehdi Rezaeian

Abstract: Vision-based robot control is a method to motion control of a robot using information extracted from visual sensors. In traditional approaches, a model of robot and camera are needed. Obtaining these models are time consuming and sometimes impossible. Recently, intelligent methods are used to cope the above challenges. In this paper, a hybrid fuzzy controller is proposed to control a robot manipulator. Visual inputs of the controller are provided by Kinect and outputs are the rotation of joints motors. The hybrid controller contains two controllers. The first controller is based on fuzzy inverse model which approximates real inverse model of robot using gathered data. In order to increase accuracy, a fuzzy expert controller is designed and it is used when the end-effector is in the predefined near-goal area. Since determining exact value of the fuzzy expert controller parameters is impossible, in addition to make system adaptive with small changes in the environment, actor-critic architecture is used. This architecture is a well known continuous

reinforcement learning methods. The proposed method is applied to control a real robot manipulator (ARM_6AX18). Experimental results show that using the proposed method in practice, the end-effector reaches from any random start position to the goal position with a good accuracy in robot workspace.

Keywords: Visual servoing, Fuzzy systems, Fuzzy inverse model, Robot manipulator, Reinforcement learning, Actor-critic.

۱- مقدمه

کنترل بینامینا^۱ عبارت است از استفاده از اطلاعات حسگرهای بصری برای کنترل ربات. این اطلاعات می‌تواند شامل نقاطی از تصویر، خط‌ها و یا ناحیه خاصی از تصویر باشند [۱]. از جمله حسگرهایی که در این زمینه از آن استفاده می‌شود دوربین است که علاوه بر اینکه اطلاعات زیادی در اختیار کنترل گر قرار می‌دهد، از نظر اقتصادی نیز مقرون به صرفه است [۲]. دوربین می‌تواند بر روی بازوی ربات^۲ یا به صورت مستقل^۳ قرار گیرد [۱]. کنترل بینامینا به سه گروه تقسیم‌بندی می‌شوند [۱]:

- روش‌های مبتنی بر موقعیت^۴
- روش‌های مبتنی بر تصویر^۵
- روش‌های ترکیبی^۶

در روش‌های مبتنی بر موقعیت، از تصویر به دست آمده برای تخمین مختصات سه‌بعدی هدف موردنظر نسبت به دوربین یا یک صفحه مختصات جهانی استفاده می‌شود. بنابراین نیاز به مدل محیط و اطلاعات دقیق از دینامیک ربات و همچنین مشخصات دوربین است. در روش‌های مبتنی بر تصویر، تلاش بر این است که خطای بین تصویر فعلی و تصویر مطلوب به حداقل برسد. در سیستم‌هایی که از روش ترکیبی استفاده شده است، از ترکیبی از دو روش قبل استفاده شده است. در این روش برخلاف روش‌های مبتنی بر موقعیت نیازی به مدل محیط نیست و همگرایی را برخلاف روش‌های مبتنی بر تصویر ضمانت می‌کند.

در این مقاله تمرکز بر روی کنترل با استفاده از روش‌های مبتنی بر تصویر است. در روش‌های سنتی کنترل مبتنی بر تصویر نیاز به دانستن مدل ربات-دوربین، رابطه بین ویژگی‌های تصویر و سینماتیک ربات، است. با استفاده از این اطلاعات در هر مرحله از فرآیند کنترل ماتریس ژاکوبین و معکوس آن برای صدور فرمان کنترلی محاسبه شود [۳]. یکی

از چالش‌های پیش‌رو در این روش معکوس‌پذیر نبودن ماتریس ژاکوبین است. علاوه بر آن برای محاسبه ماتریس ژاکوبین نیاز به دانستن پارامترهای دوربین نیز است. در [۲] از وبکم ساده برای تصویر برداری و استخراج ویژگی‌ها استفاده شده است. همچنین از شبکه‌های عصبی برای تخمین معکوس ماتریس ژاکوبین استفاده شده است. در واقع برای تخمین هر یک از سطرها ماتریس ژاکوبین که بیان‌گر نسبت تغییرات یکی از ویژگی‌های تصویر به تغییرات یکی از مفاصل‌های ربات است، یک شبکه عصبی در نظر گرفته شده است. استفاده از این روش دو مشکل ذکر شده در استفاده از ماتریس ژاکوبین را حل می‌کند. البته این روش به کنترل ربات در فضای دو بعدی می‌پردازد. علاوه بر آن روش ارائه شده توانایی تطبیق‌پذیری با تغییرات را ندارد. در [۳] از تکنیک‌های فازی برای تخمین مدل ربات-دوربین استفاده شده است. در این مقاله از کنترل گر فازی معکوس به منظور تخمین مدل معکوس ربات-دوربین برای محاسبه سرعت مفصل‌های ربات استفاده شده است. در مقاله مذکور با استفاده از یک دوربین که مستقل از ربات نصب شده است، بازوی ربات را در یک فضای دو بعدی در مسیر حلزونی شکل حرکت داده موقعیت مجری‌نهایی در هر گام حرکتی ثبت می‌شود. با استفاده از این ویژگی‌ها و مقادیر مفاصل ربات مدل فازی معکوس آموزش داده شده و مورد استفاده قرار گرفته است. دقت ناکافی و نبود معکوس برای همه نقاط کار فرآیند ضعف این دسته از روش‌ها است.

گروهی از پژوهش‌های صورت گرفته در زمینه کنترل بازوی ربات، از یادگیری تقویتی استفاده کرده‌اند. از آنجا که محیط کاری ربات گسترده است، نمی‌توان حالات و عمل‌ها را به صورت مجزا مشخص کرد. در پژوهش‌های [۴، ۵، ۶، ۷] برای رفع این مشکل به گسسته‌سازی فضای حالت و عمل پرداخته است. حالات سیستم تقسیم‌بندی‌های فضای کاری در قاب دوربین و عمل‌ها، یک سری عمل از پیش تعیین شده برای مفاصل ربات است. گسسته‌سازی همواره با مشکلاتی همراه است که از جمله این مشکلات می‌توان به تنگنای ابعاد و عدم تضمین عملکرد بهینه اشاره کرد. در پژوهش [۸] گسسته‌سازی در فضای حالت و عمل صورت نگرفته و از شبکه عصبی برای تخمین تابع ارزش حالت عمل استفاده شده است. در اینجا یادگیری تقویتی در تنظیم درست پارامترهای شبکه عصبی

^۱ Visual Servoing

^۲ Eye-in-Hand

^۳ Eye-to-Hand

^۴ Position-Based Visual Servoing

^۵ Image-Based Visual Servoing

^۶ 2 ½ D Visual Servoing

تقویتی و انواع آن می‌پردازیم. در ادامه معماری عملگر-نقاد و یادگیری تقویتی در فضای پیوسته بیان می‌شود.

۱-۲ مدل‌سازی فازی

مدلسازی فازی روشی برای نمایش دانش خبره به صورت زبانی و با استفاده از مجموعه ای از قواعد اگر-آنگاه است که این قوانین ساختار مدل را ایجاد می‌کنند. یک مدل فازی می‌تواند از ابتدا و با استفاده از داده طراحی شود. به عبارت دیگر مدل فازی نقش یک تقریب زنده جامع را برای سیستم ایفا می‌کند. قضیه تقریب زندگی جامع سیستم فازی بیان می‌کند که برای یک سیستم ناشناخته $y = F(x)$ وقتی تابع مذکور پیوسته باشد، می‌توان با جمع‌آوری داده‌ها یک سیستم فازی شامل قواعد اگر-آنگاهی طراحی کرد که با دقت دلخواه تقریبی از تابع $F(x)$ باشد [۳].

دو مدل مشهور فازی عبارتند از [۱۲]:

- مدل فازی ممدانی^۲
- مدل فازی سوگنو^۳

در این مقاله از مدل فازی سوگنو مرتبه صفر استفاده شده است. در این مدل فازی، ساختار قواعد به صورت زیر است [۳]. مرتبه این مدل فازی معادل سیستم فازی استاندارد است که حالت خاصی از سیستم‌های ممدانی است.

$$R_i = \text{If } x_1 \text{ is } A_{i1} \text{ and } \dots x_n \text{ is } A_{in} \text{ then } y_i \quad (1)$$

$$= b_i \quad i = 1, 2, \dots, k$$

در اینجا R_i بیانگر قاعده i ام، $[x_1, \dots, x_n]$ ورودی‌های سیستم فازی، $[A_{i1}, \dots, A_{in}]$ مجموعه‌های فازی در نظر گرفته شده برای مقادیر ورودی، و y_i خروجی هر قاعده است. خروجی سیستم فازی از رابطه (۲) محاسبه می‌شود.

$$\hat{y} = \frac{\sum_{i=1}^K \beta_i y_i}{\sum_{i=1}^K \beta_i} \quad (2)$$

که در اینجا β_i شدت آتش هر قاعده است که:

$$\beta_i = \prod_{j=1}^n \mu_{A_{ij}}(x_j) \quad i = 1, 2, \dots, k \quad (3)$$

که $\mu_{A_{ij}}(x_j): R \rightarrow [0, 1]$ درجه تعلق ورودی x_j به مجموعه فازی A_{ij} در مقدم قاعده R_i است.

نقش موثری دارد. هرچند یادگیری تقویتی دو قابلیت یادگیری تنها با یک معیار عددی راندمان و آموزش برخط را داراست، اما سرعت یادگیری آن پایین است و نیاز به تکرارهای زیاد آزمایش دارد.

در این مقاله یک کنترلگر فازی ترکیبی ارائه می‌شود. در این کنترلگر از کنترلگر فازی معکوس برای رساندن معجری نهایی ربات پیرامون هدف استفاده می‌شود و یک کنترلگر خبره فازی برای رساندن دقیق معجری نهایی به هدف بکار گرفته شده است. سپس پارامترهای این کنترلگر خبره فازی به منظور بهبود کارایی و تطبیق پذیری با تغییرات جزئی با روش یادگیری تقویتی تنظیم می‌شوند.

در اینجا از کینکت^۱ [۹] بعنوان حسگر بصری برای استخراج ویژگی‌های سه بعدی و اطلاعات عمق بهره برده می‌شود. مزیت استفاده از کینکت این است که علاوه بر اطلاعات تصویر RGB، اطلاعات عمق را نیز در اختیار قرار می‌دهد و نیازی به استفاده از چند دوربین برای استخراج اطلاعات عمق، که خود نیاز به دانستن پارامترهای داخلی و خارجی دوربین‌ها است، نمی‌باشد. روش ارائه شده بر روی بازوی ربات ARM-6AX18 [۱۰]، به کار گرفته شده است. همچنین برای گرفتن اطلاعات تصویر کینکت XBOX-360 [۱۱] مورد استفاده قرار گرفته است. دوربین جدا از ربات و به صورت مستقل نصب شده است.

به طور کلی سهم علمی این تحقیق شامل موارد زیر است:

- ارائه ساختار کنترلگر ترکیبی فازی برای حل مسئله کنترل بازوی ربات به صورت ترکیبی از کنترلگر معکوس فازی و کنترلگر خبره فازی.
- ارائه راهکاری برای تنظیم برخط کنترلگر خبره فازی با استفاده از یادگیری تقویتی
- حل چالش‌های به کارگیری روش‌های فوق در یک مسئله عملی و به کارگیری آن‌ها بر روی بازوی ربات واقعی

ساختار مقاله به شرح زیر است که در بخش دوم مفاهیم پایه می‌آید. در بخش سوم روش پیشنهادی شرح داده می‌شود. در بخش چهارم نتایج آزمایشات بر روی بازوی ربات ذکر شده است. در نهایت در بخش پنجم نتیجه گیری و کارهای آینده بیان شده است.

۲- مفاهیم پایه

در این بخش ابتدا به معرفی مدل سازی فازی، مدل معکوس فازی و مفاهیم مرتبط با آن پرداخته می‌شود. سپس به بیان مفاهیم پایه یادگیری

^۲ Mamdani Fuzzy Model

^۳ Takagi-Sugeno Fuzzy Model

^۱ Kinect

$$U = G(x(k), x(k + 1)) \quad (۶)$$

به این معنی که عمل یکنمایی مثل U وجود دارد که حالت سیستم را از $x(k)$ به $x(k + 1)$ می‌برد. برای به دست آوردن این مدل معکوس، $x(k)$ و $x(k + 1)$ ورودی سیستم، و U به عنوان خروجی سیستم در نظر گرفته می‌شود. سیستم فازی نهایی در نهایت معکوس فرآیند را تخمین می‌زند.

۳-۲ یادگیری تقویتی

یادگیری تقویتی یک روش یادگیری بر خط می‌باشد که بصورت تعاملی و با استفاده از پاداش و جریمه، عملکرد مطلوب به عامل آموزش داده می‌شود. این پاداش یا جریمه را سیگنال تقویتی می‌نامند. به بیان دیگر یادگیری از طریق حداکثر کردن سیگنال تقویتی که مقداری عددی^۱ است، صورت می‌گیرد [۱۴، ۱۵]. روش‌های یادگیری تقویتی بر اساس نحوه تخمین تابع ارزش شکل گرفته‌اند. متداول‌ترین آنها روش‌هایی هستند که از خطای تفاضل موقتی^۲ استفاده می‌نمایند.

۴-۲ یادگیری تفاضل موقتی

ساده‌ترین روش تفاضل موقتی^۳ که آن را $TD(0)$ می‌نامند، برای تخمین تابع ارزش حالت بصورت زیر بکار می‌رود [۱۵، ۱۶]:

$$V(s_t) \leftarrow V(s_t) + \alpha_t [r_{t+1} + \gamma V(s_{t+1}) - V(s_t)] \quad (۷)$$

در این رابطه $0 < \alpha_t < 1$ نرخ یادگیری^۴ و $0 < \gamma < 1$ ضریب نزول^۵ و مقدار پاداش آنی هستند. s_t و s_{t+1} به ترتیب بیان‌کننده حالت سیستم در زمان t و $t + 1$ و V مشخص‌کننده ارزش حالت است. در این رابطه، عبارت داخل کروشه را خطای تفاضل موقتی می‌نامند. دو معماری معروف استفاده شده برای کاربرد یادگیری تقویتی در فضای پیوسته عملگر-نقاد^۶ و نقاد-تنها^۷ هستند. در ادامه روش عملگر-نقاد که در این مقاله از آن استفاده شده است، شرح داده می‌شود.

قواعد فازی را می‌توان با استفاده از داده‌ها استخراج کرد. به عبارت دیگر اگر بردار X ورودی سیستم در هر مرحله و Y خروجی سیستم ناشناخته باشد، یعنی:

$$X^T = [x_1, \dots, x_p], \quad Y^T = [y_1, \dots, y_n] \quad (۴)$$

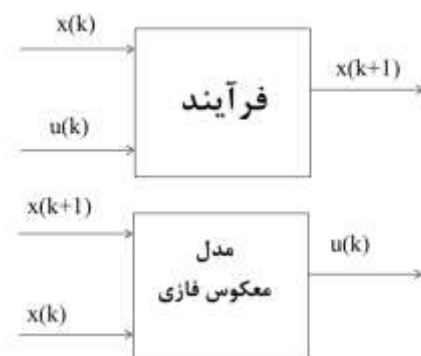
همانطور که در مقالات مختلف از جمله [۱۳] بیان شده است خوشه بندی یکی از راه‌های شناخته شده در تولید قواعد فازی با استفاده از داده‌های ورودی خروجی است. این روش داده‌ها را به گروه‌های همگن تقسیم بندی می‌کند و این کار باعث می‌شود که داده‌های مشابه که رفتار مشابهی در سیستم دارند در یک گروه قرار گرفته و کل خوشه حاصل، نماینده آن داده‌ها باشد. سپس برای این گروه از داده‌ها قاعده ای در نظر گرفته خواهد شد. در این مقاله برای ساخت مدل فازی از جعبه ابزار فازی متلب و دستور genfis3 استفاده شده است. این دستور از فازی C-means برای خوشه‌بندی داده‌ها و تولید قواعد فازی استفاده می‌شود.

۲-۲ مدل معکوس

برای به دست آوردن مدل معکوس و تخمین تابع f^{-1} می‌توان از مدل فازی معکوس یا شبکه‌های عصبی استفاده کرد. در اینجا از مدل فازی معکوس استفاده شده است که با استفاده از داده جمع شده ساخته می‌شود. به این صورت که اگر تمام حالات فرآیند مشخص باشد، داریم [۷]:

$$x(k + 1) = f(x(k), u(k)) \quad (۵)$$

که $x(k + 1)$ حالت فرآیند در زمان $k + 1$ ، $x(k)$ حالت فرآیند در زمان k و $u(k)$ فرمان کنترلی در زمان k است. بنابراین



شکل ۱: فرآیند و مدل فازی معکوس آن

معادله دینامیکی معکوس این فرآیند به صورت (۶) است:

^۱ Scalar

^۲ TD-Error

^۳ Temporal Difference

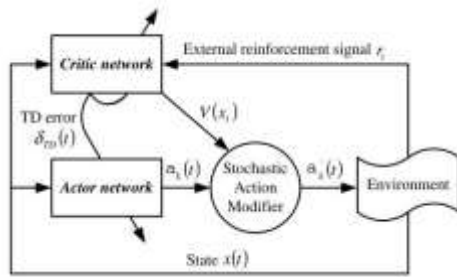
^۴ Learning Rate

^۵ Discount Factor

^۶ Actor-Critic

^۷ Critic-Only

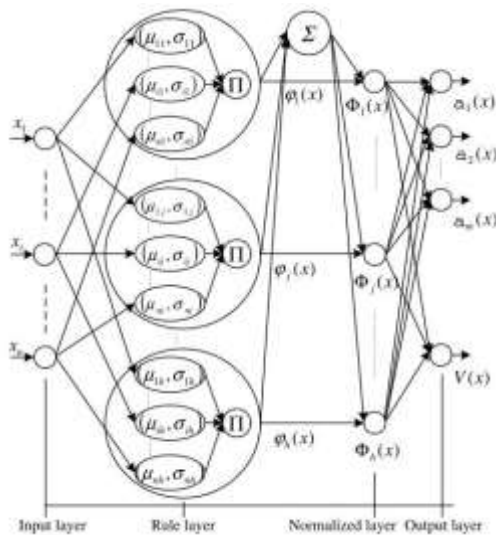
۵-۲ روش عملگر-نقاد



شکل ۲: معماری یک سیستم کنترلی FACRLN [۱۷]

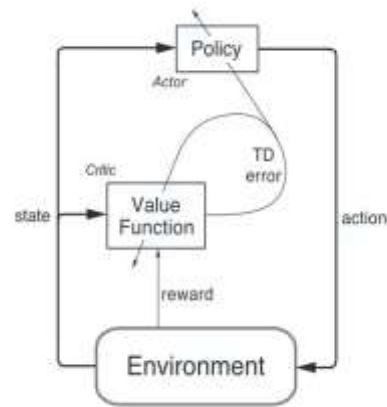
پیشنهادی را ارائه می‌دهد. سپس اصلاح کننده عمل تصادفی^۴، مقدار تصادفی با توجه به میزان ارزش آن $V(x_t)$ و عمل انتخابی $a_k(x_t)$ تولید کرده و با اضافه کردن آن به عمل انتخابی، عمل واقعی را تولید می‌کند. شبکه نقاد سیگنال تقویتی را دریافت کرده و خطای تفاضل موقت $(\delta_{TD}(t))$ را تولید می‌کند و به شبکه عملگر می‌فرستد.

شکل ۴ این شبکه را نمایش می‌دهد. از آنجایی که دو شبکه نقاد و عملگر ورودی یکسانی دارند و فقط در خروجی دارای تفاوت هستند، لایه های یک تا سه برای آنها مشترک است. در ادامه به توضیح هر کدام از این لایه ها می‌پردازیم. لایه اول همان لایه ورودی است که بردار ورودی را دریافت می‌کند و به لایه بعد ارسال می‌کند. لایه دوم لایه قواعد است که هر واحد آن قسمت مقدم یکی از قواعد را با n تابع عضویت گوسی شکل نشان می‌دهد. خروجی این لایه شدت آتش هر قاعده است که از رابطه (λ) به دست می‌آید.



شکل ۴: یک شبکه FACRLN [۱۷]

روش عملگر-نقاد از جمله روش های یادگیری تفاضل موقت است که ساختاری مستقل از تابع ارزش برای بیان سیاست گذاری دارد. این ساختار سیاست گذاری را به دلیل انتخاب عمل، عملگر می‌نامند. تخمین زنده تابع ارزش که نقش نقدکننده عمل صورت گرفته را بر عهده دارد، راه نقاد می‌گویند. یادگیری در این روش به صورت برسیاست^۱ است. نقاد در حال یادگیری و انتقاد از سیاستی است که عملگر در حال اجرای آن است. انتقاد به فرم خطای تفاضل موقت^۲ برای یادگیری عملگر و نقاد استفاده می‌شود. شکل ۲ این فرآیند را نشان می‌دهد [۱۵].



شکل ۳: معماری عملگر-نقاد [۱۳]

۶-۲ شبکه یادگیری عملگر-نقاد فازی

در [۱۷] ثابت شده است که سیستم استنتاج فازی در واقع یک تقریب زنده است که می‌تواند هر مجموعه داد ورودی-خروجی را تقریب زند. یک سیستم فازی دانش بشر را به گونه ای قابل فهم کد می‌کند ولی در یادگیری و تطبیق پذیری خوب عمل نمی‌کند. از طرف دیگر شبکه های عصبی توانایی یادگیری و تحمل خطا را دارند ولی برای نمایش دانش مناسب نیستند. بنابراین می‌توان از خصوصیات هر دو این تقریب زنده ها استفاده کرد.

با توجه به مطالب ذکر شده، شبکه های یادگیری عملگر-نقاد فازی را که از این به بعد به اختصار آنها را FACRLN^۳ می‌نامیم، معرفی می‌شود. معماری این شبکه در شکل ۳ نشان داده شده است. دو بخش مهم این معماری شبکه نقاد و عملگر هستند. شبکه عملگر، نگاشت بین حالت-عمل را یاد می‌گیرد و شبکه نقاد، تابع ارزش سیاست دنبال شده، با توجه به ورودی سیستم را آموزش می‌بیند [۱۷]. شبکه عملگر، عمل

^۱ On Policy

^۲ TD-Error

^۳ Fuzzy Actor-Critic Reinforcement Learning Network

^۴ Stochastic Action Modifier

می آید، حرکت مفاصل دیگر از طریق یک موتور صورت می گیرد. نحوه صدور فرمان به موتورهای این ربات از طریق تعیین زاویه چرخش آنها است. همچنین این امکان فراهم شده است که در صورت لزوم بتوان زوایای هر یک از موتورها را از طریق بازخوردی که موتورها به ما می دهند، به دست آورد. هدف از طراحی این کنترل گر، رساندن مجری نهایی بازوی ربات، از هر نقطه شروع اولیه ای به نقطه هدف تعیین شده در محدوده کاری ربات است. به منظور کنترل بازو از یک کنترل گر ترکیبی فازی استفاده شده است که شامل کنترل گر معکوس فازی و کنترل گر خبره فازی است. کنترل گر ترکیبی در حالتی که محیط بدون تغییر باشد، یعنی ربات و دوربین جای ثابتی داشته باشند، کارایی دارد. در صورت ایجاد تغییر در محیط دیگر این کنترل گر دقت لازم را ندارد. علاوه بر آن تعیین دقیق پارامترهای کنترل گر خبره فازی امکان پذیر نیست و ضعف هایی دارد. لذا از یادگیری تقویتی به منظور یادگیری تغییرات به صورت برخط و برطرف کردن ضعف های کنترل گر خبره فازی استفاده شده است.

۳-۱ کنترل گر فازی ترکیبی

برای طراحی کنترل گر ترکیبی فازی از دو کنترل گر استفاده شده است. کنترل گر اول نقش رساندن مجری نهایی به محدوده نزدیک هدف را به عهده دارد و یک کنترل گر معکوس فازی است. کنترل گر دوم، کنترل گر خبره فازی است که با استفاده از دانش خبره طراحی شده است. این کنترل گر در ناحیه نزدیک هدف، کنترل ربات را به دست می گیرد. دیاگرام کنترلی کنترل گر ترکیبی در شکل ۵ نمایش داده شده است. همانطور که در شکل مشاهده می کنید، برای فرآیند کنترل، هر دو کنترل گر مختصات مجری نهایی در تصویر احتیاج دارند. با توجه به اینکه در اینجا هدف کنترل بازو در فضای سه بعدی است، علاوه بر مختصات مجری نهایی در تصویر، به اطلاعات عمق باید دسترسی داشت. با توجه به این خاصیت کینکت که هر دو تصویر RGB و عمق را در فرام می کند، این اطلاعات در اختیار کنترل گر قرار گرفته می شود. در نهایت مختصات مجری نهایی با سه پارامتر x ، y و Z مشخص می شود و به کنترل گر داده می شود. x ، y مختصات مجری نهایی در تصویر RBG گرفته شده توسط دوربین و Z مقدار متناظر با نقطه X و y مشخص شده در قسمت قبل، در تصویر عمق گرفته شده از کینکت است. علاوه بر این کنترل گر فازی معکوس سه ورودی دیگر دارد که همان مقادیر زاویه موتورها هستند. مقادیر موتورها در واقع مشخص کننده حالت فعلی سیستم هستند.

۳-۱-۱ کنترل گر معکوس فازی

$$\varphi_j(x_t) = \exp\left(-\sum_{i=1}^n \frac{(x_{it} - \mu_{ij})^2}{2\sigma_{ij}^2}\right) \quad (8)$$

که μ و σ میانگین و انحراف معیار هر کدام از توابع عضویت هستند و x_{it} برابر با μ_{ij} متغیر ورودی بردار x_t است. لایه سوم خروجی لایه دوم را نرمال سازی می کند و لایه چهارم خروجی شبکه عملگر و نقاد است. خروجی نقاد حاصل تخمینی از ارزش حالت است. به عبارت دیگر نگاهی از یک حالت سیستم به یک ارزش نقاد است. عملگر نقش یک انتخاب کننده عمل را اجرا می کند که نگاهی از آنچه سیستم از فضای n بعدی درک می کند به فضای m بعدی عمل ها است. خروجی عملگر و نقاد $V(x_t)$ به صورت زیر محاسبه می شود:

$$a_i(x_t) = \sum_{j=1}^h w_{ij} \Phi_j(x_t) \quad i = 1, \dots, m \quad (9)$$

$$V(x_t) = \sum_{j=1}^h v_j \Phi_j(x_t) \quad (10)$$

در (۹)، $\Phi_j(x_t)$ شدت آتش نرمال شده قاعده j ام و w_{ij} وزن های بین خروجی نرمال شده قاعده j ام و خروجی عملگر i ام می باشد. در (۱۰)، v_j وزن بین خروجی نرمال شده قاعده j ام و بخش نقاد است. به منظور داشتن کاوش در محیط عمل محاسبه شده به طور مستقیم در محیط اعمال نشده و نویز گوسی n_k به آن اضافه می شود که پهنای گوسی رابطه عکس با ارزش حالت فعلی ربات دارد. $a_i(x_t)$ عمل محاسبه شده بعد از اضافه کردن نویز به عمل محاسبه شده است.

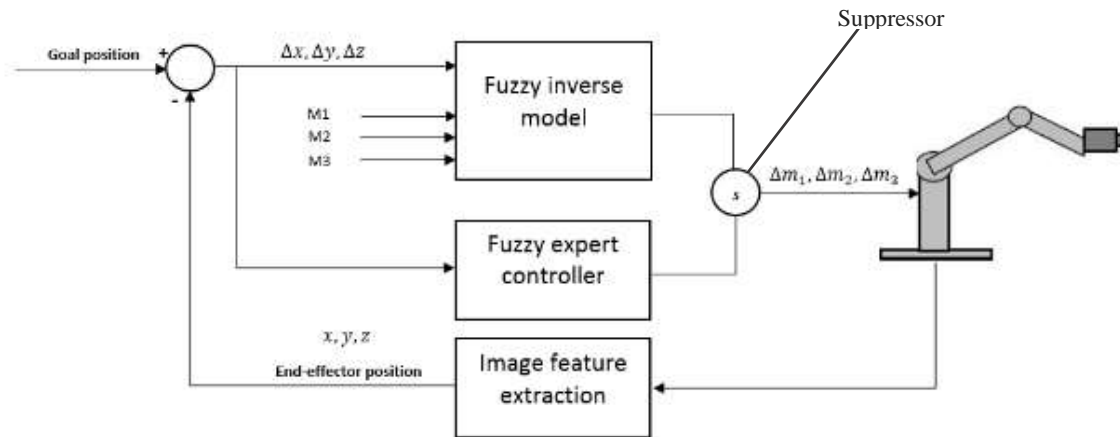
$$a'_i(x_t) = a_i(x_t) + n(0, \sigma_v(t)) \quad (11)$$

$$\sigma_v(t) = \frac{1}{1 + \exp(2V(x_t))} \quad (12)$$

$n(0, \sigma_v(t))$ یک مقدار متغیر تصادفی با توزیع گوسی با میانگین صفر و انحراف معیار $\sigma_v(t)$ تولید می کند که به عمل محاسبه شده اضافه می شود. اگر مقدار $V(x_t)$ ، یعنی ارزش حالت کوچک باشد مقدار کاوش زیاد است و بالعکس.

۳- روش ارائه شده

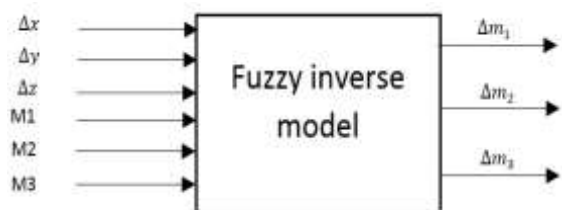
در این فصل، به معرفی روش ارائه شده برای کنترل بازوی ربات با سه درجه آزادی می پردازیم. این ربات دارای سه مفصل پایه، شانه و آرنج است که به غیر از مفصل شانه که توسط دو موتور به حرکت در



شکل ۵: کنترل گر ترکیبی فازی

[۱۹،۱۸] دو روش رایج برای این کار جدول مراجعه^۱ و خوشه‌بندی است. در اینجا برای تولید سیستم فازی از روش خوشه‌بندی که در بخش قبل به آن اشاره شد استفاده شده است. از آنجا که حجم داده‌های جمع‌آوری شده از محیط زیاد بود، امکان تولید قواعد به صورت دستی و استفاده از تقسیم بندی مشبک و روش‌های دیگر نبود. براساس آنچه در [۲۰] آمده است خوشه‌بندی یکی از روش‌های مرسوم در کاهش حجم داده‌ها است. در نتیجه داده‌ها را خوشه‌بندی کرده و برای هر خوشه یک قاعده فازی در نظر گرفته شده است. آزمایشات اولیه نشان داد، زمانی که تنها از کنترل گر معکوس فازی استفاده می‌شود دقت لازم را ندارد. نتایج آزمایش‌های صورت گرفته در بخش ۰ و جدول ۴ آورده شده است. این عدم دقت به دلیل معکوس پذیر نبودن سیستم در برخی نقاط و همچنین نویز موجود در داده‌های جمع‌آوری شده می‌باشد. علاوه بر این در طراحی کنترلگر معکوس، در حین ساخت آن هدف حداقل کردن خطای خروجی کنترلگر تخمین زده شده u^{\wedge} ، و مدل معکوس واقعی یعنی u است. در حالی که در عمل و استفاده از این مدل، هدف حداقل کردن فاصله مجری نهایی y^{\wedge} و هدف y است. با توجه به آنچه گفته شد و به دلیل حساسیت ۱۰۰٪ کنترل کننده معکوس به خطای مدل سازی، این کنترل کننده به تنهایی برای کنترل دقیق بازوی ربات کارایی ندارد. به همین دلیل در کنار آن از یک کنترل کننده دیگر که توسط خبره طراحی شده، استفاده شده است. وجود کنترل کننده معکوس فازی باعث می‌شود که فضای عملکرد کنترل کننده خبره تنها محدود به نقاط اطراف نقطه کار شده و در نتیجه ساختار و طراحی آن توسط خبره با مشاهده عملکرد واقعی سیستم ساده‌تر گردد. لذا به منظور کنترل نهایی، کنترل گر

کنترل گر معکوس فازی نهایی یک سیستم فازی سوگنو مرتبه صفر است که دارای شش ورودی و سه خروجی است (-شکل ۶). سه ورودی $[Vx, Vy, Vz]$ فاصله مجری نهایی از هدف در تصاویر RGB و عمق است. سه ورودی دیگر یعنی $[m1, m2, m3]$ مقادیر سه موتور مفاصل ربات هستند که بیان کننده وضعیت فعلی ربات است. همانطور که قبل تر به آن اشاره شد، یکی از راه‌های طراحی یک سیستم فازی، بکارگیری داده‌های از پیش جمع‌آوری شده است. به این صورت که ورودی و خروجی های یک فرآیند را مشاهده، و ذخیره کرده و از آنها در تخمین تابعی که تقریبی از معکوس فرآیند اصلی باشد، استفاده کنیم [۳].



شکل ۶: مدل فازی معکوس

رای جمع‌آوری اطلاعات در این مورد خاص ربات را در موقعیت متفاوت نسبت به مبدأ آن قرار می‌دهیم و مختصات مجری نهایی و مقادیر موتورها را ذخیره می‌کنیم. موقعیت‌های قرارگیری مجری نهایی باید به گونه‌ای باشند که تمام فضای کاری ربات را پوشش دهد. به عنوان مثال در اینجا زوایای یکی از سه موتور ربات را در هر مرحله از جمع‌آوری اطلاعات ده درجه تغییر داد. بعد از جمع‌آوری اطلاعات نوبت به تولید سیستم فازی متناسب با این داده‌های ورودی-خروجی است. روش‌های مختلفی برای تولید سیستم فازی از روی این داده‌ها وجود دارد. بر اساس

^۱ Table Look-Up

سیستم فازی که به عنوان کنترل گر خبره فازی در بخش ۳-۱-۲ معرفی کردیم را به صورت دانش اولیه در نظر می‌گیریم. در شکل ۴ هر کدام از واحدهای لایه قواعد یکی از قواعد کنترل گر خبره فازی را نمایش می‌دهند. همچنین وزن‌هایی که از لایه سوم خارج شده و وارد بخش عملگر لایه چهارم می‌شود، به همان خروجی های کنترل گر خبره فازی مقاداردهی شده است. آنچه در حین فرآیند آموزش تغییر می‌کند، وزن‌های خارج شده از لایه سوم شبکه به لایه چهارم برای هر دو بخش نقاد و عملگر است. این نکته قابل توجه است که این شبکه جایگزین کنترل گر خبره فازی می‌شود و فقط در نواحی نزدیک هدف مورد استفاده قرار می‌گیرد.

فرآیند آموزش در این روش به این صورت است که ابتدا موقعیت مجری نهایی و فاصله آن با هدف را توسط دوربین کینکت به دست آورده و فرمان کنترلی صادر می‌شود. اگر فرمان صادر شده مناسب بود، یعنی فاصله مجری نهایی تا هدف کم شده باشد، سیستم پاداش می‌گیرد و در غیر این صورت جریمه می‌شود. بعد از محاسبه عمل و مقدار ارزش حالت طبق رابطه (۱۴) و (۱۵) برای آموزش سیستم لازم است وزن‌های بخش نقاد و عملگر بروزرسانی شود. به منظور محاسبه میزان بروزرسانی وزن‌ها ابتدا باید خطای تفاضل موقتی^۲ محاسبه شود [۱۴].

$$\delta_{TD}(t) = r_t + \gamma V(x_{t+1}) - V(x_t) \quad (13)$$

x_t و x_{t+1} مشخص کننده حالت سیستم در زمان‌های t و $t+1$ است. r_t سیگنال تقویتی و γ ضریب کاهش^۳ و V در اینجا مشخص کننده ارزش هر حالت است. خطای تفاضل موقتی اگر مثبت باشد نشان‌دهنده این است که حالت فعلی با توجه به عمل انجام شده بهتر از حالت قبلی بوده و اگر منفی باشد بالعکس. از این خطای تفاضل موقت برای به روز رسانی وزن‌های بخش عملگر و نقاد استفاده می‌شود. وزن‌های بخش عملگر به صورت زیر بروزرسانی می‌شوند:

$$w_{ij}(t+1) = w_{ij}(t) + \alpha_A \delta_{TD}(t) \frac{a'_i(x_t) - a_i(x_t)}{\sigma_v(t)} \Phi_j(x_t) \quad (14)$$

نحوه بروزرسانی وزن‌های بخش نقاد نیز به صورت زیر است:

$$v_i(t+1) = v_i(t) + \alpha_c \delta_{TD}(t) \Phi_j(x_t) \quad (15)$$

معکوس، مجری نهایی را به نواحی اطراف هدف رسانده و در نواحی نزدیک هدف از کنترل گر فازی خبره، استفاده شده است.

۳-۱-۲ کنترل گر خبره فازی

کنترل گر خبره فازی توسط خبره و با توجه به مشاهدات صورت گرفته در نحوه عملکرد ربات در محیط طراحی شده است. این کنترل گر هم از نوع سوگنو مرتبه صفر است و همانند کنترل گر معکوس فازی، دارای سه ورودی است. ورودی‌ها به ترتیب بیان‌کننده فاصله مجری نهایی تا هدف تعیین شده در بعد x ، y و z هستند. در اینجا چون کنترل گر در فضای محدودی مورد استفاده قرار می‌گیرد، بدون نیاز به حالت سیستم و مقادیر موتورها هم عملکرد سیستم قابل قبول است. لذا به منظور کاهش پیچیدگی از آن‌ها استفاده نشده است. خروجی نیز مانند کنترل گر معکوس فازی میزان تغییر در زوایای موتورهای هریک از مفاصل است. این کنترل گر زمانی که مجری نهایی در محدوده نزدیک هدف قرار می‌گیرد، خروجی کنترل گر معکوس فازی را غیرفعال کرده و خود، کنترل ربات را بر عهده می‌گیرد. محدوده نزدیک عبارت است از ناحیه ای اطراف هدف، که از قبل تعیین شده است.

۳-۲ تنظیم بر خط کنترل گر فازی ترکیبی

اگر در محیط تغییری ایجاد شود، مثلاً ربات نسبت به دوربین جابجا شود، و یا زاویه دوربین تغییر کند، دیگر کنترل گر ترکیبی دقت مناسب را ندارد. علاوه بر آن به دلیل طراحی کنترل گر خبره فازی توسط دانش خبره، و تنظیم نبودن دقیق پارامترهای آن، این کنترل گر دارای ضعف‌هایی از جمله حرکات نوسانی اطراف هدف می‌باشد. روش‌های مختلفی همچون مجموع مربعات خطا تکرارشونده^۱ و یادگیری تقویتی برای تنظیم پارامترهای کنترل گر وجود دارد. در این پژوهش، از معماری عملگر-نقاد برای تنظیم پارامترهای کنترل گر خبره فازی که یکی از روش‌های یادگیری تقویتی فازی است، استفاده شده است.

نکته قابل توجه دیگر آن این است که هرچند محیط عملکرد کوچک باشد، با این حال با توجه به اینکه یادگیری تقویتی از اطلاعات کمی استفاده می‌کند و تنها معیار یادگیری یک مقدار عددی است، زمان آموزش طولانی دارد. مورد دیگر اینکه زمانی که فرآیند آموزش طولانی باشد در عمل باعث داغ شدن موتورهای ربات شده و آزمایشات عملی را با مشکلاتی همراه خواهد کرد. لذا در این پژوهش به منظور افزایش سرعت یادگیری، دانش اولیه به سیستم تزریق می‌کنیم. بدین صورت که

^۲ Temporal Difference

^۳ Discount Factor

^۱ Recursive Least Square Error

هایی در آزمایشات استفاده شده است، XBOX-360 [۱۵] است. محل قرارگیری کینکت در پشت سر ربات و در جایی ثابت در محیط است (شکل ۹).



شکل ۹: نحوه قرارگیری کینکت نسبت به ربات
جدول ۱: مشخصات موتورهای ربات

نوع موتور	مفصل	تعداد موتور	راستای حرکت	زاویه چرخش	شمار موتور	دقت موتور
پایه	AX-18	۱	افقی	۳۰۰	۱	۰،۲۹
شانه	AX-18	۲	عمودی	۱۸۰	۲،۳	۰،۲۹
آرنج	MX-28	۱	عمودی	۳۰۰	۴	۰،۰۰۸

۴-۱ جمع آوری داده برای کنترل گر معکوس فازی و طراحی آن

به منظور جمع آوری اطلاعات باید ربات را در موقعیت های مختلف در محیط کاری آن قرار داد و اطلاعات مربوطه موقعیت مجری نهایی و مقادیر موتورها را جمع آوری کرد. در این مقاله موقعیت مجری نهایی و مقادیر موتورها با تغییر پنج درجه در مقادیر موتورها در هر مرحله از جمع آوری اطلاعات ذخیره شده است. این فرآیند تا زمانی که کل ناحیه در دسترس ربات پیموده شود ادامه داده می شود. برای مشخص کردن مجری نهایی در تصویر و به دست آوردن مختصات آن، یکی از گیره های مجری نهایی را به رنگ قرمز درآورده شده است. در هر بار گرفتن

در روابط (14) و (15) $\Phi_j(x_t)$ ، α_A و α_C به ترتیب شدت آتش نرمال شده، نرخ آموزش پارامترهای عملگر و نرخ آموزش پارامترهای نقاد می باشند. در هر بار اجرای عملی در محیط، مقادیر وزن های عملگر و نقاد بروزرسانی می شوند. در یادگیری تقویتی هدف حداکثر کردن پاداش در بلند مدت است و با توجه به آنچه گفته شد، در تکرارهای متوالی سیستم به سمتی پیش می رود که به حداکثر پاداش که همان رسیدن به هدف با کمترین زمان است، برسد.

۴- آزمایش ها

در این پژوهش از بازوی ربات ARM-6AX18 [۵] (شکل ۷) با سه درجه آزادی استفاده شده است. این ربات دارای سه مفصل پایه، شانه و آرنج است که در شکل ۸ نمایش داده شده است. در جدول ۱ مشخصات موتورهای بازوی ربات نشان داده شده است. موتورهای شماره دو و سه دو طرف شانه ربات قرار گرفته و حرکت عکس یکدیگر را دارند. فرمان کنترلی میزان تغییرات زوایای هر یک از موتورها است که توسط کنترل گر صادر شده و به آن اعمال می شود.



شکل ۷: بازوی ربات ARM-6AX18



شکل ۸: مفصلات ربات

در ادامه به توضیح نحوه جمع آوری اطلاعات برای ساخت کنترل گر معکوس فازی خواهیم پرداخت. کینکتی که برای تعیین مختصات مجری

مجری نهایی در محدوده $-100 < \nabla x < 100$ ، $-100 < \nabla y < 100$ و $-300 < \nabla z < 300$ از هدف قرار گرفته شده باشد.

۴-۲ طراحی کنترل گر خبره فازی

کنترل گر با استفاده از مشاهداتی که از عملکرد سیستم در عمل به دست آمده، طراحی شده است. این کنترل گر یک سیستم فازی سوگنو مرتبه صفر است. در طراحی این کنترل گر فرض شده که موتور پایه برای رساندن مجری نهایی در بعد، x موتورهای شانه در بعد y و موتور آرنج برای بعد z به طور مستقل عمل می کنند. توابع عضویت ورودی کنترل گر خبره فازی طراحی شده در شکل ۱۱ نمایش داده شده است. برای هر سه ورودی سه تابع عضویت گوسی شکل به نام های Zero، Neg، و Pos در نظر گرفته شده است. سیستم فازی مذکور سه خروجی دارد که هر کدام مشخص کننده تغییرات در زوایای هر کدام از مفصل های ربات است. خروجی های این سیستم برای موتور پایه دارای مقادیر $[۱۵؛ ۰؛ -۲۰]$ ، موتور شانه $[۲۰؛ ۰؛ -۶۰]$ و موتور آرنج $[۶۰؛ ۰؛ -۶۰]$ هستند. برای این سیستم ۲۷ قاعده نوشته شده است. برخی قواعد کنترل گر خبره فازی در جدول ۳ نشان داده شده است.

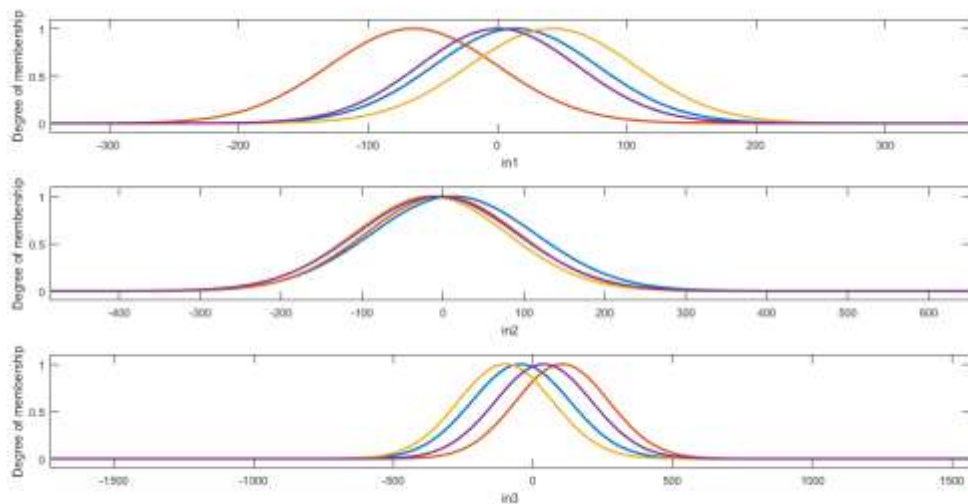
۴-۱ نتایج آزمایش کنترل گر ترکیبی فازی

زمانی که از کنترل گر ترکیبی استفاده می شود، مجری نهایی با دقت قابل قبولی به هدف تعیین شده می رسد. برای آزمایش چندین نقطه هدف در محیط کاری ربات در نظر گرفته شده است. با استفاده از کنترل گر معکوس فازی، و کنترل گر ترکیبی سعی شده تا مجری نهایی به هدف تعیین شده برسد.

تصویر توسط کینکت نقاط قرمز رنگ را در تصویر RGB پیدا کرده و مرکز آنها را به عنوان مختصات فعلی مجری نهایی در نظر گرفته می شود. سپس نقطه متناظر با این نقطه را، در تصویر عمقی را به عنوان عمق مجری نهایی در نظر گرفته می شود.

پس از جمع آوری اطلاعات نوبت به طراحی مدل فازی معکوس می رسد. ورودی-خروجی های مدل همان چیزی است که در شکل ۶ نشان داده شده است. از آنجایی که مقادیر موتورهای شانه مکمل هم هستند، تنها یکی از آنها در طراحی مدل استفاده شده است. برای مقدار $m3$ هم زاویه موتور آرنج در نظر گرفته شده است. برای ساخت مدل فازی از جعبه ابزار فازی متلب استفاده شده است. دستور genfis3 مجموعه داده های ورودی و خروجی را به صورت جداگانه دریافت می کند. علاوه بر آن نوع سیستم فازی که در اینجا "Sugeno" است را دریافت می کند. خروجی این تابع یک سیستم فازی سوگنو مرتبه صفر است که همان مدل فازی معکوس است که با استفاده از خوشه بندی داده ها به وجود آمده است. شکل ۱۰ توابع عضویت سه ورودی کنترل گر فازی معکوس که مربوط به مقادیر مجری نهایی در سه بعد، بعد از خوشه بندی را نشان می دهد. همچنین جدول ۲ برخی قواعد فازی تولید شده توسط دستور genfis3 را نشان می دهد.

گفته شد که در آزمایش های عملی کنترل گر معکوس به تنهایی دقت لازم را ندارد و در ناحیه نزدیک هدف از کنترل گر خبره فازی استفاده شده است. در اینجا ناحیه نزدیک هدف به موقعیتی گفته می شود که



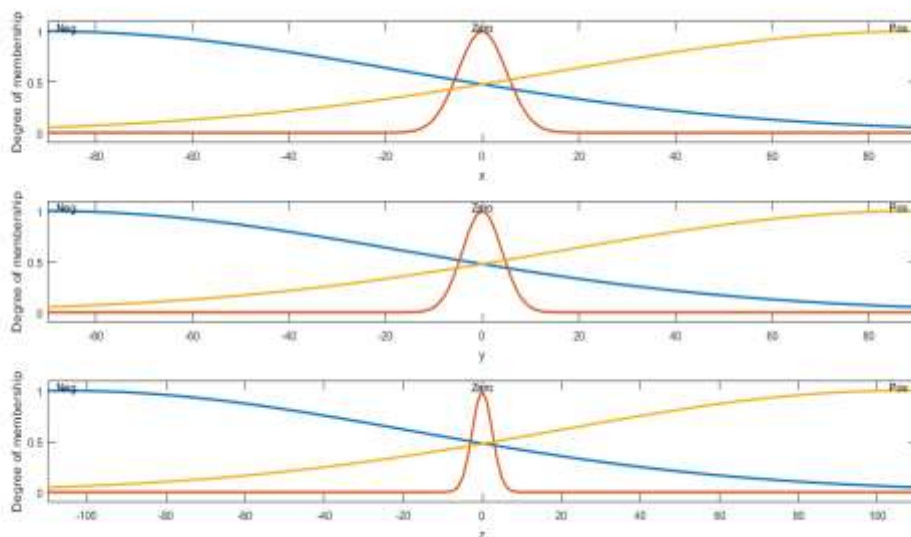
شکل ۱۰: توابع عضویت سه ورودی کنترل گر معکوس

جدول ۲: برخی قواعد کنترل گر معکوس فازی

شماره قاعده	قاعده
۱	If (in1 is in1c1) and (in2 is in2c1) and (in3 is in3c1) and (in4 is in4c1) and (in5 is in5c1) and (in6 is in6c1) then (out1 is out1c1)(out2 is out2c1)(out3 is out3c1)
۲	If (in1 is in1c2) and (in2 is in2c2) and (in3 is in3c2) and (in4 is in4c2) and (in5 is in5c2) and (in6 is in6c2) then (out1 is out1c2)(out2 is out2c2)(out3 is out3c2)
۳	If (in1 is in1c3) and (in2 is in2c3) and (in3 is in3c3) and (in4 is in4c3) and (in5 is in5c3) and (in6 is in6c3) then (out1 is out1c3)(out2 is out2c3)(out3 is out3c3)

جدول ۳: برخی قواعد کنترل گر خبره فازی

شماره قاعده	قاعده
۱	If (x is Neg) and (y is Neg) and (z is Neg) then ($\Delta m1$ equals -15)($\Delta m2$ equals -20)($\Delta m3$ equals 60)
۲	If (x is Neg) and (y is Neg) and (z is Zero) then ($\Delta m1$ equals -15)($\Delta m2$ equals -20)($\Delta m3$ equals 0)
۳	If (x is Neg) and (y is Neg) and (z is Pos) then ($\Delta m1$ equals -15)($\Delta m2$ equals -20)($\Delta m3$ equals -60)
۴	If (x is Neg) and (y is Zero) and (z is Neg) then ($\Delta m1$ equals -15)($\Delta m2$ equals 0)($\Delta m3$ equals 60)
۵	If (x is Neg) and (y is Zero) and (z is Zero) then ($\Delta m1$ equals -15)($\Delta m2$ equals 0)($\Delta m3$ equals 0)



شکل ۱۱: توابع عضویت ورودی کنترل گر خبره فازی

آن محدوده قرار گیرد هدف در دسترس آن قرار دارد. در آزمایش های انجام شده این محدوده به صورت $[-30,0]$ ، $[-30,30]$ و $[-10,30]$ در ابعاد x, y, z و تعریف شده است.

جدول ۴ خطای سیستم در زمان استفاده از کنترل گر معکوس فازی به تنهایی و کنترل گر ترکیبی فازی در پنج نقطه مختلف هدف، را با هم مقایسه می کند. برای تشخیص رسیدن مجری نهایی به هدف ناحیه محدودی اطراف هدف در نظر گرفته شده است که اگر مجری نهایی در

است. بدیهی است که همانطور که گفته شد، کنترل گر ترکیبی دیگر دقت لازم را ندارد.

بعد از اعمال تغییرات در محیط حال نوبت به آموزش سیستم می‌رسد. آموزش سیستم مانند آنچه است که در بخش ۳-۲ آورده باشد. عمل پیشنهادی و نحوه به روزرسانی وزن‌ها بر اساس روابط (۱۱)، (۱۴) و (۱۵) صورت می‌گیرد.

مورد دیگری اینکه سیگنال تقویتی برای محاسبه خطای تفاضل موقت مورد نیاز است. در آزمایشات صورت گرفته سیگنال تقویتی در نظر گرفته شده به صورت رابطه (۱۶) است.

$$\begin{cases} 1 & \text{reach goal} \\ -1 & \text{lose goal} \\ 0.1 & \text{dis}_{pre} > \text{dis}_{cur} \\ -0.1 & \text{dis}_{pre} < \text{dis}_{cur} \\ -0.01 & \text{dis}_{pre} = \text{dis}_{cur} \end{cases} \quad (16)$$

منظور از dis_{pre} و dis_{cur} به ترتیب فاصله اقلیدسی مجری نهایی از هدف در گام قبلی و فعلی است.

در روابط بروزرسانی وزن های عملگر و نقاد دو پارامتر نرخ آموزش $0 < \sigma_A < 1$ و $0 < \sigma_C < 1$ مطرح شد. در آزمایش های انجام شده این پارامترها به صورت $\sigma_C = 0.01$ و $\sigma_A = 0.08$ در نظر گرفته شده است. این مقادیر اگر خیلی کوچک باشد، فرآیند آموزش کند می شود و در مقابل اگر مقدار بزرگی انتخاب شود، ممکن است منجر به نوسان پیرامون جواب بهینه شود.

فرض شده است، زمانی که سیستم در طی فرآیند آموزش به جایی رسید که در ۲۰ مرحله^۱ پی در پی به هدف رسید، فرآیند آموزش متوقف شد. با توجه به این فرض، سیستم را ۴۶۷ مرحله، آموزش طول کشیده شده است. از این تعداد ۳۰۲ مرحله موفق، و ۱۶۵ مرحله با شکست مواجه شد.

همانطور که در جدول ۴ نمایش داده شده، کنترل گر ترکیبی خطا را کاهش داده و به هدف می‌رساند. منظور از خطا فاصله مجری هایی تا محدوده هدف است. شکل ۱۲ موقعیت مجری نهایی را در زمانی که به هدف رسیده و خطا صفر شده را نشان می‌دهد.



شکل ۱۲: موقعیت مجری نهایی در زمان رسیدن به هدف

جدول ۴: مقایسه دقت دو کنترل گر معکوس و ترکیبی

شماره آزمایش	کنترل گر	خطای بعد $(px)x$	خطای بعد $(px)y$	خطای بعد z (mm)
۱	معکوس	-۲۰	-۲۳	۰
	ترکیبی	reach		
۲	معکوس	۰	۳۴	۰
	ترکیبی	reach		
۳	معکوس	۰	۶۳	۱۰۲
	ترکیبی	reach		
۴	معکوس	۵۶	۴۹	-۲۵
	ترکیبی	reach		
۵	معکوس	-۳	۸۲	۱۲۵
	ترکیبی	reach		

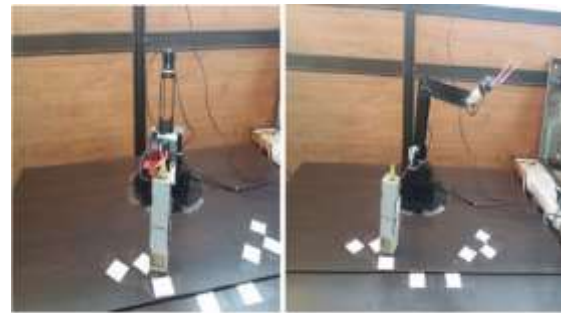
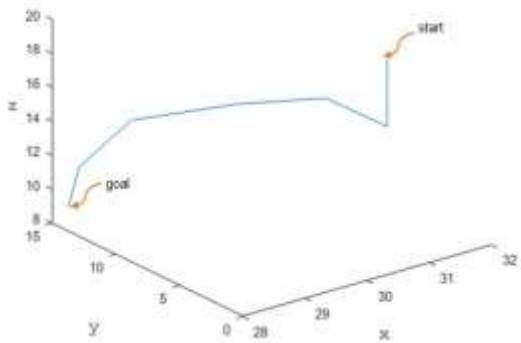
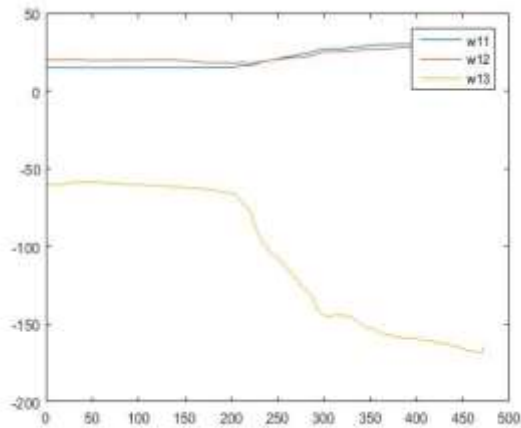
۴-۲ تنظیم بر خط کنترل گر خیره فازی

در بخش ۳-۲ مطرح شد که از یادگیری تقویتی و روش عملگر-نقاد برای آموزش برخط تغییرات به سیستم و همچنین بهبود عملکرد کنترل گر خیره فازی استفاده شده است. به منظور یادگیری تغییرات و برای انجام آزمایش ها در این زمینه، ابتدا نیاز است که تغییراتی در سیستم و محیط کاری ربات ایجاد شود. لذا در اینجا تغییری که در محیط ایجاد شده، تغییر ده سانتی متر در موقعیت میزی است که ربات در آن قرار دارد. میز به اندازه ده سانتی متر از دیوار فاصله گرفته و جلوتر قرار گرفته

^۱ Episode

۳-۴ تست کنترل گر آموزش دیده به صورت برخط

زمانی که به نظر رسید که سیستم به اندازه کافی آموزش دیده است، فرآیند آموزش متوقف و برای شش نقطه از پیش تعیین شده برای تست، سیستم مورد ارزیابی قرار گرفت. جدول ۵ نتایج حاصله از این تست را قبل و بعد از آموزش نشان می دهد. اگر مجری نهایی به محدوده تعریف شده برای هدف رسیده باشد، خطا صفر در نظر گرفته شده و عبارت reach به جای خطا نوشته شده است. شکل ۱۳ تصویر نقطه شروع حرکت و توقف مجری نهایی در یکی از آزمایش ها ذکر شده در جدول ۵ را نمایش می دهد.



شکل ۱۳: موقعیت اولیه و پایانی مجری نهایی

جدول ۵: مقایسه عملکرد کنترل گر قبل و بعد از آموزش

شماره آزمایش	خطای کنترل گر قبل از آموزش	خطای کنترل گر آموزش شده
۱	[-6,22,14]	reach
۲	[0,29,13]	reach
۳	[-9,19,18]	reach
۴	[-4,-22,14]	reach
۵	[-19,-27,21]	reach
۶	[5,-3,26]	reach

شکل ۱۵: موقعیت مجری نهایی در یکی از آزمایشات

۵- نتیجه گیری

در این مقاله یک کنترل گر ترکیبی فازی برای رساندن مجری نهایی بازوی ربات به هدف ارائه شد. این کنترل گر شامل یک کنترل گر معکوس فازی و یک کنترل گر خبره فازی است. کنترل گر معکوس فازی برای رساندن مجری نهایی به اطراف هدف استفاده شد. آزمایش ها نشان داد که این کنترل گر به دلیل معکوس پذیر نبودن برخی نقاط کاری ربات و خطای موجود در جمع آوری اطلاعات، به تنهایی دقت کافی را دارا نیست. لذا برای رفع این مشکل از کنترل گر خبره فازی در نواحی نزدیک هدف استفاده شد. در ادامه از معماری عملگر-نقاد یادگیری تقویتی برای تنظیم برخط کنترل گر جهت تنظیم بهتر پارامترهای کنترل گر خبره فازی و نیز قابلیت تطبیق پذیری سیستم استفاده شد.

نتایج آزمایش های صورت گرفته نشان داد روش مذکور عملکرد مناسبی دارد و ترکیب معماری عملگر-نقاد و کنترل گر معکوس فازی می تواند منجر به این شود که این دو ساختار ضعف های یکدیگر را بپوشانند. از یک طرف با استفاده از کنترل گر معکوس فازی توانستیم

همانطور که گفته شد، وزن های عملگر ابتدا به خروجی سیستم مقادیری می شود و در طی فرآیند آموزش تغییر می کنند. با استفاده از مقادیر جدید وزن ها پس از آموزش، سیستم خود را با تغییرات اعمال شده به محیط مطابقت دهد. شکل ۱۴ نمودار تغییرات وزن های مربوط به قاعده اول را نشان می دهد. در شکل مذکور ابتدا قاعده نشان داده شده، کمتر تحریک شده، در نتیجه تغییرات آن کم است. در ادامه روند آموزش این قاعده بیشتر قاعده غالب بوده و در نتیجه تغییرات وزن های آن بیشتر شده است. در [۲۲،۲۱] فیلم نتیجه آزمایشات صورت گرفته برای کنترل گر ترکیبی و تطبیق پذیر قرار داده شده است. شکل ۱۵ موقعیت مجری نهایی در هر گام حرکت در یک نمونه آزمایش، از حالت اولیه تا رسیدن به هدف را نشان می دهد.

- [8] Z. Miljkovic, M. Mitic, M. Lazarevic, and B. Babic, "Neural Network Reinforcement Learning for Visual Control of Robot Manipulators," *Expert Systems with Applications*, vol. 40, pp. 1721–1736, 2013.
- [9] M. Deisenroth, C. Rasmussen, and D. Fox, "Learning to Control a Low-Cost Manipulator Using Data-Efficient Reinforcement Learning," *International Conference on Robotics: Science & Systems*, pp. 57–64, 2011.
- [10] Robotic Arms. (n.d.). Pishrobot. [Online]. Available: http://www.pishrobot.com/en/products/robotic_arms.htm. Accessed 19 Aug 2016.
- [11] Kinect | Xbox 360. Xbox.com. (n.d.). [Online]. Available: <http://www.xbox.com/en-US/xbox-360/accessories/kinect>. Accessed 19 Aug 2016.
- [12] J. Jang, C. Sun, and E. Mizutani, *Neuro-fuzzy and Soft Computing*. Upper Saddle River, NJ: Prentice Hall, 1997.
- [13] S. Guillaume. "Designing Fuzzy Inference Systems from Data: An Interpretability-Oriented Review." *IEEE Transactions on fuzzy systems*, pp. 426-443, 2001.
- [14] R. S. Sutton and A. G. Barto, "Introduction to Reinforcement Learning," *IEEE Transactions on Robotics and Automation*, MIT Press, 1998.
- [15] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement Learning: A survey," *Journal of Artificial Intelligence Research*, vol. 1, no. 1, pp. 237–285, 1996.
- [16] R. Sutton, and A. G. Barto, "Reinforcement learning," *Journal of Cognitive Neuroscience*, vol. 11, no. 1, pp. 126-134, 1999.
- [17] X. S. Wang, Y. H. Cheng, and J. Q. Yi "A Fuzzy Actor-Critic Reinforcement Learning Network," *Journal of Information Sciences*, vol. 177, pp. 3764–3781, 2007.
- [18] L. X. Wang, (1997): *A course in fuzzy systems and control*. 1. Aufl. Upper Saddle River, NJ: Prentice Hall PTR.
- [19] R. Babuska, *Fuzzy Modeling for Control*. Boston, MA: Kluwer Academic, 1998.
- [20] FJ. Chang, and YT. Chang. "Adaptive Neuro-Fuzzy Inference System for Prediction of Water Level in Reservoir." *Advances in Water Resources*. pp. 1-10, 2006.
- [21] Fuzzy hybrid control of robot with camera, aparat, 2016. [Online]. Available: <http://www.aparat.com/v/5UWeh>. Accessed: 08 Jul 2016.
- [22] Fuzzy adaptive control of robot with camera, aparat, 2016. [Online]. Available: <http://www.aparat.com/v/blqN0>. Accessed: 17 Oct 2016.
- فضای آموزش و محدوده جستجو را در یادگیری تقویتی کاهش داده و لذا سرعت یادگیری را افزایش دهیم. لازم به ذکر است یک ضعف روشهای یادگیری تقویتی نیاز به تکرار زیاد تجربیات است. رفع این مشکل بخصوص در هنگام انجام آزمایش با یک ربات واقعی (نه مدل شبیه سازی شده) که انجام آزمایشات زمانبر و پرهزینه است یک مزیت مهم به حساب می آید. از طرف دیگر استفاده از یادگیری تقویتی امکان تنظیم برخط پارامترهای کنترل گر فازی و تطبیق پذیری آن را فراهم می کند.
- از آنجایی که برای گرفتن اطلاعات بصری از کینکت استفاده شده، و دوربین کینکت در فضای باز قابل استفاده نیست، لذا برای مسیر آینده این پژوهش مناسب است که از دوربین معمولی و به صورت استریو، برای گرفتن اطلاعات از محیط استفاده شود تا بتوان از این سیستم در فضای باز نیز استفاده کرد.

۶- مراجع

- [1] D. Kragic, and H. Christensen, "Survey on Visual Servoing for Manipulation," *Computational Vision and Active Perception Laboratory*, Fiskartorpsv 15, 2002.
- [2] F. Nadi, "Visual Servoing Control of Robot Manipulator with Jacobian Matrix Estimation," (in Persian) M.S. Thesis, Faculty of Electrical and Computer Engineering, Yazd University 2014.
- [3] P. Goncalves, L. Mendonca, J. Sousaand, and J. Pinto, "Uncalibrated Eye-to-Hand Visual Servoing Using Inverse Fuzzy Models," *IEEE Transactions on Fuzzy Systems*, pp. 341–353, 2008.
- [4] C. Distanto, A. Anglani, and F. Taurisano, "Target Reaching by Using Visual Information and Q-learning Controllers," *Autonomous Robots*, vol. 9, pp. 41–50, 2000.
- [5] A. Anglani, F. Taurisano, R. De Giuseppe, C. Distanto, and L. Lecce, "Learning to Grasp by Using Visual Information Robot System and Controller Architecture," *Autonomous Robots*, vol. 9, pp. 41–50, 2000.
- [6] M. Sadeghzadeh, "Self-Learning Visual Servoing of Robot Manipulator Using Explanation-Based Fuzzy Neural Networks and Q-Learning," Ph.D. Dissertation, University of Guelph, 2014.
- [7] K. Shibata, M. Sugisaka, and K. Ito, "Hand Reaching Movement Acquired Through Reinforcement Learning," in *Proceedings of 2000 KACC (Korea Automatic Control Conference)*, 2000.