

# مسیریابی دو هدفه مبتنی بر منطق فازی برای ربات‌های متحرک در محیط‌های پویای ناشناخته خلوت

سمیه مغاری، جعفر فتحعلی و زهرا براتی

دانشکده علوم ریاضی، دانشگاه صنعتی شاهرود، شاهرود، ایران

تاریخ پذیرش: ۱۴۰۰/۸/۱۴

تاریخ دریافت: ۱۴۰۰/۴/۱۶

## چکیده

در سال‌های اخیر، استفاده از ربات‌ها برای کارهای خدماتی و انجام عملیات در محیط‌های پرخطر مورد استقبال گسترده قرار گرفته است. یکی از کارهای پیچیده در این عملیات، اجتناب ربات از برخورد با موانع غیرقابل پیش‌بینی در هنگام حرکت از مبدأ به مقصد است. این پژوهش، یک الگوریتم مسیریابی دو هدفه مبتنی بر منطق فازی برای ربات‌های متحرک در محیط‌های پویای ناشناخته را ارائه می‌دهد. اهداف این الگوریتم کمینه‌سازی زمان رسیدن از مبدأ به مقصد و همچنین کمینه‌سازی مسافت طی شده است. نتایج شبیه‌سازی در نرم‌افزار Matlab نشان می‌دهد که هنگامی که محیط حرکت خلوت باشد، الگوریتم پیشنهادی از کارایی بالایی برخوردار است.

## ۱ سرآغاز

امروزه از ربات‌ها برای کارهای خدماتی و انجام عملیات در محیط‌های پرخطر به صورت گسترده استفاده می‌شود [۲، ۶، ۹]. یکی از کارهای نسبتاً پیچیده در کنترل خودکار ربات‌ها، اجتناب از برخورد با موانع مختلف و غیرقابل پیش‌بینی در هنگام حرکت

---

عبارات و کلمات کلیدی: کنترل‌کننده فازی، برنامه ریزی حرکت ربات، مسیریابی چندهدفه، محیط پویای ناشناخته خلوت

در یک محیط ناشناخته و پویا است [۱۳]. در اینجا، ناشناخته بودن محیط به این معناست که ربات نقشه محیط را ندارد (برخلاف روش‌هایی مانند [۱۷]) و در هر گام، از اطلاعات ورودی حسگرها برای مسیریابی استفاده می‌کند. همچنین پویا بودن محیط به معنای حضور موانع متحرک در آنجا است [۱۶]. ربات، در زمان حرکت در این محیط‌ها فعالیت‌های مختلفی را برای اجتناب از برخورد با موانع انجام می‌دهد. دریافت اطلاعات محیط توسط حسگرها، پردازش و پالایش اطلاعات حسگرها برای تشخیص موانع و مسیرهای باز مابین آنها (حفره‌ها)، تصمیم‌گیری و انتخاب یک حفره برای عبور و همچنین ارسال دستور به واحدهای مکانیکی برای ایست، چرخش، حرکت و تنظیم سرعت، فعالیت‌های اصلی ربات برای مسیریابی و اجتناب از موانع است [۸، ۱۳، ۱۴، ۱۸، ۲۰]. این فعالیت‌ها معمولاً در مسیریابی خودکار وسایل بدون سرنشین نیز انجام می‌شود [۷]. در این پژوهش، ما بر روی استراتژی حرکت شامل تشخیص زمان‌های حرکت و ایست و همچنین اولویت‌بندی و انتخاب حفره‌های باز مابین موانع در حین مسیریابی تمرکز داریم و اهداف بهینه‌سازی شامل کمینه بودن طول مسیر نهایی و زمان رسیدن ربات به مقصد است.

مسائل برنامه‌ریزی مسیر ربات‌های متحرک از سه دیدگاه نوع، زمان و محیط دسته بندی می‌شوند [۱]. از دیدگاه نوع، این مسائل به دو دسته سراسری و محلی تقسیم می‌شوند. در مسائل از نوع سراسری، نقشه کل محیط در دسترس است و برنامه‌ریزی مسیر پیش از حرکت ربات امکان‌پذیر است اما در مسائل از نوع محلی، نقشه در دسترس نیست و اطلاعات محیط توسط حسگرهای ربات در زمان حرکت جمع‌آوری و پردازش می‌شود. مسئله مورد بررسی در این پژوهش از نوع محلی است که به آن مسیریابی در محیط‌های ناشناخته گفته می‌شود. از دیدگاه زمان، این مسائل به دو دسته برخط و غیربرخط تقسیم می‌شوند. در نوع برخط، برنامه‌ریزی مسیر در زمان حرکت ربات انجام می‌شود اما در نوع غیربرخط، برنامه‌ریزی مسیر پیش از حرکت ربات انجام می‌شود. در این مقاله یک مسئله از نوع برخط مورد پژوهش قرار گرفته است. از دیدگاه محیط، مسائل در دو دسته محیط ایستا و محیط پویا مورد بررسی قرار می‌گیرند. در محیط‌های ایستا، موانع همواره بدون حرکت یا تغییر شکل هستند اما در محیط‌های پویا، برخی از موانع جابجا می‌شوند یا تغییر شکل می‌دهند. در این پژوهش، برنامه‌ریزی مسیر ربات‌ها

در محیط‌های پویا مدنظر قرار گرفته است.

برای برنامه‌ریزی مسیر ربات‌های متحرک در محیط‌های ناشناخته و پویا، الگوریتم‌های مختلفی توسعه داده شده است. مهناتی<sup>۱</sup> و دی‌وانگ<sup>۲</sup> در [۱۰] یک الگوریتم برای اجتناب از موانع و کمینه‌سازی طول مسیر طی شده توسط ربات در محیط‌های نامشخص، ارائه دادند. آنها با روش فراابتکاری ازدحام ذرات، علاوه بر بهینه‌سازی تابع هدف، انتخاب مسیرهای هموار و اجتناب از پیچ‌های تند در مسیر را در الگوریتم خود پشتیبانی کردند تا ربات همواره مسیر امن برای رسیدن به مقصد را بییماید. البته آنها پویایی محیط را در الگوریتم پیشنهادی در نظر نگرفتند. اوریول<sup>۳</sup> و موران<sup>۴</sup> در [۱۵] یک الگوریتم مبتنی بر روش فراابتکاری کلونی مورچگان برای اجتناب ربات از مانع در محیط‌های ناشناخته دارای موانع پیچیده ارائه دادند. آنها موانع را با شکل‌های محدب و غیرمحدب در نظر گرفتند؛ اما موانع پویا نبودند. رحمانیار<sup>۵</sup> و راخمانیا<sup>۶</sup> نیز یک الگوریتم ژنتیک برای اجتناب ربات از مانع در زمان مسیریابی در محیط‌های شامل موانع چندگانه ارائه دادند [۱۱] که محیط، کاملاً ناشناخته در نظر گرفته نشده بود و کروموزوم‌ها یک مسیر ممکن برای رسیدن به هدف بودند. آنها از روش ترکیب معمول که شکستن کروموزوم‌ها و تعویض نیمه‌ها است استفاده کردند و سپس حفره‌های مابین موانع به عنوان نقاط جهش به کروموزوم‌ها اضافه می‌شد. جی<sup>۷</sup> و کوئی<sup>۸</sup> در [۵] از روش میدان‌های پتانسیل برای برنامه‌ریزی حرکت ربات‌ها در محیط ناشناخته استفاده کردند. در این روش ربات به عنوان نقطه‌ای داخل میدان‌های پتانسیل در نظر گرفته می‌شود و با ترکیب کشش به سمت هدف و دافعه از موانع، مسیر پیمایش ربات محاسبه می‌شود. این الگوریتم‌ها ممکن است در بهینه‌های محلی میدان‌های پتانسیل به دام بیفتند و موفق به پیدا کردن مسیر نشوند. چن<sup>۹</sup> و همکارانش در [۳] برای هدایت ربات‌ها در محیط‌های پویای ناشناخته از تکنیک

<sup>1</sup>Mohanty

<sup>2</sup>Dewang

<sup>3</sup>Uriol

<sup>4</sup>Moran

<sup>5</sup>Rahmaniar

<sup>6</sup>Rakhmania

<sup>7</sup>Ge

<sup>8</sup>Cui

<sup>9</sup>Chen

پویایی عصبی بهره بردند تا یکی از اشکالات روش‌های مبتنی بر شبکه عصبی که امکان گیر افتادن در بهینه محلی است را حل کنند. این رویکرد، بر مبنای الهام از پویایی عصبی در موجودات زنده است که اعصاب می‌توانند مستقل از بافت‌های دیگر حرکت کنند. بنابراین، شبکه عصبی مصنوعی مانند یک شبکه توپولوژیکی پویا در نظر گرفته می‌شود که در زمان حرکت و مسیریابی ربات، قابلیت اصلاح و ارتقاء دارد. در روش پیشنهادی آنها، ربات قابلیت تغییر مسیر را داشت اما امکان توقف برای عبور موانع متحرک پیش‌بینی نشده بود. علمی<sup>۱۰</sup> و ایفی<sup>۱۱</sup> برای مسیریابی ربات در محیط‌های ناشناخته پویا، الگوریتمی مبتنی بر روش فراابتکاری ملخ ارائه دادند [۴]. آنها از رفتار جمعیت ملخ‌ها برای عبور از موانع پیش رو با انتخاب بهترین حفره مابین آنها استفاده کردند. الگوریتم ۱ ساختار کلی مسیریابی ربات‌ها در محیط‌های ناشناخته پویا با روش فراابتکاری ملخ را نمایش می‌دهد. در این الگوریتم، هنگامی که ربات با مانع روبرو می‌شود، جمعیتی از ملخ‌ها در اطراف مانع تولید می‌شود و پس از تعدادی تکرار و جابجایی ملخ‌ها، بهترین آنها بر اساس تابع برازش انتخاب می‌شود و جایگاه آن به عنوان مقصد موقتی بعدی ربات در نظر گرفته می‌شود. این الگوریتم به دلیل استفاده از رویکرد مبتنی بر جمعیت، در هنگام رویارویی با مانع باید حجم زیادی از پردازش را انجام دهد. همچنین برای محیط‌های پویا، راهبرد مؤثری برای استفاده از تکنیک توقف و تبدیل شدن به مانع ایستا ندارد.

(الف) مقداردهی اولیه ربات، مقصد و موقعیت مانع

(ب) حرکت ربات به سمت مقصد تا زمانی که مانع را تشخیص ندهد

(ج) اگر مانعی در مسیر وجود دارد، الگوریتم ملخ فعال شود:

۱. تولید جمعیت ملخ به صورت تصادفی

۲. انتخاب ملخ با بهترین مقدار تابع هدف در بین جمعیت

۳. حرکت ربات به سمت موقعیت بهترین ملخ کنونی

(د) اگر ربات به مقصد نرسیده، برو به مرحله (ب)

<sup>10</sup>Elmi

<sup>11</sup>Efe

(ه) پایان.

یکی از تکنیک‌های پرکاربرد در الگوریتم‌های مسیریابی در محیط‌های ناشناخته پویا، استفاده از کنترل‌کننده فازی برای تصمیم‌گیری پیرامون انتخاب یک مسیر از بین چند مسیر ممکن در گام‌های مختلف پیمایش است [۲، ۱۲، ۱۳، ۱۹]. در این الگوریتم‌ها فرض بر این است که ربات همواره از مختصات مکان فعلی خود و مختصات مقصد آگاهی دارد و همچنین از تعدادی حسگر برای پویش موانع مجاور خود استفاده می‌کند. حسگرها اطلاعات فاصله با موانع و حفره‌های مابین آنها را در اختیار پردازنده قرار می‌دهند تا بتواند با استفاده از این اطلاعات در روال‌های پردازشی خود، یکی از حفره‌های مابین موانع را برای عبور انتخاب کند. کنترل‌کننده فازی عملیات اولویت‌بندی و انتخاب حفره برای عبور را انجام می‌دهد. پس از انتخاب حفره، فرمان چرخش و حرکت به واحدهای مکانیکی ربات داده می‌شود و سپس ربات تا نقطه موردنظر جابجا می‌شود. این کارها تا زمانی که ربات به مقصد برسد، تکرار می‌شود. منطق فازی امکان مدیریت ابهام و نقص در اطلاعات را فراهم می‌کند و به همین علت در مسیریابی ربات‌ها در محیط‌های ناشناخته برای رویارویی با عدم قطعیت مورد استفاده قرار می‌گیرد. یکی از مزیت‌های مهمی که کنترل‌کننده‌های فازی فراهم می‌کنند، امکان مدل‌سازی و اضافه کردن استنتاج‌های شبیه مغز انسان به الگوریتم‌ها است. به عنوان مثال، هنگامی که انسانی در یک محیط ناشناخته به سمت هدفی مشخص در حال حرکت است، سعی می‌کند موانع را به گونه‌ای پشت سر بگذارد که از مسیر اصلی به سمت هدف منحرف نشود. همین رویکرد به سادگی در یک کنترل‌کننده فازی قابل مدل‌سازی است.

کنترل‌کننده فازی مختلف می‌توانند ورودی‌ها و خروجی‌های متفاوت داشته باشند و همچنین قواعد استنتاج آنها با هم فرق داشته باشد. روسو<sup>۱۲</sup> و همکارانش برای مسیریابی ربات‌ها در محیط‌های داخل ساختمان یک کنترل‌کننده فازی-عصبی طراحی کردند [۱۲] که براساس یک سلسله‌مراتب دو سطحی از رفتارهای ربات توسعه داده شده بود. رفتارهای سطح بالا مانند خروج از بن‌بست یا رفتن به مقصد، از یک یا چند رفتار سطح پایین یا اولیه مانند چرخش، دنبال کردن دیواره یا حرکت مستقیم به یک مختصات

<sup>12</sup>Rusu

تشکیل شده بودند و کنترل‌کننده فازی با استفاده از اطلاعات حسگرها، حالت‌های رفتاری را تشخیص و توصیه می‌کرد. ژائو<sup>۱۳</sup> و لی<sup>۱۴</sup> در [۱۹] برای مسیریابی گروهی ربات‌ها در محیط‌های ناشناخته پویا روشی ارائه دادند که از دو کنترل‌کننده فازی برای اجتناب از مانع و مسیریابی استفاده می‌کند. رویکرد کلی روش آنها در الگوریتم ۱ نمایش داده شده است. در هر گام از این الگوریتم، ابتدا محیط پویا می‌شود. سپس در صورت وجود خطر برخورد، کنترل‌کننده اجتناب از مانع بر مبنای میزان خطر برخورد با مانع، سرعت چرخ‌های راست و چپ ربات را تنظیم می‌کند و برای جلوگیری از برخورد، مسیر حرکت ربات را تغییر می‌دهد. در زمانی که خطر برخورد با مانع ربات را تهدید نمی‌کند، کنترل‌کننده دیگر که برای مسیریابی به سمت مقصد است، سرعت چرخ‌ها را براساس سرعت فعلی آنها و میزان انحراف از مسیر اصلی محاسبه می‌کند.

(الف) پویا محیط و اندازه‌گیری خطر

(ب) در صورت وجود مانع در فاصله کمتر از حد آستانه، ناوبری با کنترل‌کننده فازی اجتناب از مانع

(ج) در صورت عدم وجود مانع در فاصله کمتر از حد آستانه، ناوبری با کنترل‌کننده هدایت به مقصد

(د) اگر ربات به مقصد نرسیده، برو به مرحله (الف)

(ه) پایان.

در این مقاله، یک استراتژی مسیریابی و اجتناب از مانع در محیط‌های ناشناخته پویا براساس ترکیبی از حرکت، ایست و تغییر جهت و همچنین استنتاج مبتنی بر کنترل‌کننده فازی پیشنهادی می‌شود و سپس نتایج پیاده‌سازی و اجرا در نرم‌افزار Matlab و مقایسه با الگوریتم‌های [۱۹] و [۴] به عنوان دو الگوریتم جدید که نزدیک‌ترین فرضیات را با این پژوهش دارند، مورد بحث قرار می‌گیرد. در این راستا، بخش ۲ به تعریف مسئله و بیان شرایط محیط اختصاص یافته است. در بخش ۳، الگوریتم پیشنهادی در قالب راهبرد کلی حرکت و همچنین مشخصات کنترل‌کننده فازی پیشنهادی تشریح می‌شود.

<sup>13</sup>Zhao

<sup>14</sup>Lee

بخش ۴ نتایج شبیه‌سازی مورد بررسی قرار می‌گیرد و بخش ۵ نیز شامل خلاصه‌ای از این پژوهش و نتایج آن و همچنین پیشنهاد کارهای آتی است.

## ۲ تعریف مسئله

در این بخش، تعریف موردنیاز و فرضیات مسئله بیان می‌شود.

**تعریف ۱.۰۲.** محدوده، یک مستطیل به طول  $L_e$  و عرض  $W_e$  است که ربات‌ها و موانع داخل آن تعریف می‌شوند و پیرامون آن یک حصار غیرقابل نفوذ قرار گرفته است.

**تعریف ۲.۰۲.** در اینجا بدون از دست دادن کلیت مسئله، هر ربات را بصورت مربعی با ضلع یک متر در نظر می‌گیریم. هر ربات یک زاویه دید  $180^\circ$  درجه با مرکزیت مسیر در حال حرکت دارد و مسیر اولیه (پیش‌فرض) برای هر ربات، از مبدأ به سمت مقصد است. فرض بر این است که برای هر ربات، حداقل یک مسیر از مبدأ به مقصد وجود دارد که پیمایش بدون برخورد در این مسیر، امکان‌پذیر است. اطلاعات ورودی هر ربات محدود به مختصات مکان فعلی خود ربات، مختصات مقصد و مختصات حدود موانع قابل رؤیت است که از طریق حسگرها تخمین زده می‌شود. مجموعه کل ربات‌ها را با  $R$  نشان می‌دهیم.

**تعریف ۳.۰۲.** مانع ایستا، یک چند ضلعی غیرشفاف و غیرقابل نفوذ است که مختصات رئوس آن در طول اجرای الگوریتم تغییر نمی‌کند.

**تعریف ۴.۰۲.** مانع پویا، یک چند ضلعی غیرشفاف و غیرقابل نفوذ که مختصات رئوس آن در طول اجرای الگوریتم متغیر است. قابل ذکر است که ربات‌های متحرک برای هم مانع پویا محسوب می‌شوند.

در اینجا، ربات نقشه کل مسیر را ندارد و در هر گام، از اطلاعات محلی و مسیریابی برای رسیدن به مقصد استفاده می‌کند. در واقع ربات همواره مختصات مکان فعلی خود و مختصات مقصد را دارد و همچنین از تعدادی حسگر برای رؤیت موانع مجاور خود استفاده می‌کند. حسگرها این اطلاعات را در اختیار پردازنده قرار می‌دهند و ربات در هر گام باید یکی از حفره‌های مابین موانع را برای عبور انتخاب کند.

محیط پیمایش، یک مستطیل بزرگ شامل تعدادی مانع ایستا است که برای سادگی مسئله، آنها را چند ضلعی محدب فرض می‌کنیم. تعدادی ربات که هر کدام مبدأ و مقصد مجزایی دارند، بصورت مستقل (بدون تعامل با یکدیگر) و همه با الگوریتم یکسان درحال مسیریابی هستند و برای هر ربات، سایر ربات‌ها مانع متحرک محسوب می‌شوند. هدف هر ربات، طی کردن یک مسیر بدون برخورد از مبدأ به مقصد است. همچنین اهداف بهینه‌سازی، کمینه کردن مسیر طی شده ربات‌ها و همچنین کمینه کردن زمان رسیدن به مقصد است. همچنین فرض می‌کنیم که محیط خلوت است؛ یعنی تعداد ربات‌ها در مقایسه با مساحت محدوده، کم باشد.

### ۳ الگوریتم پیشنهادی

از آنجا که همه ربات‌ها دارای راهبرد مسیریابی یکسان و کنترل‌کننده‌های مشابه هستند، در این بخش، راهبرد مسیریابی یک ربات در هنگام مواجه شدن با موانع ایستا و پویا و همچنین انتخاب حفره‌های مابین موانع برای عبور را شرح می‌دهیم.

#### ۱.۳ راهبرد کلی حرکت

الگوریتم ۱.۳ راهبرد هر ربات برای حرکت از مبدأ به مقصد را نشان می‌دهد. این الگوریتم شامل دو بخش اصلی "پویش و تصمیم‌گیری" و "حرکت" است.

+ پویش و تصمیم‌گیری: هنگامی که ربات در مسیر خود یک مانع متحرک را در فاصله کمتر از  $D_T$  مشاهده می‌کند، یک زمان تصادفی بین ۱ تا ۳ ثانیه صبر می‌کند و عملیات پویش را تکرار می‌کند. تصادفی بودن زمان توقف برای این است که اگر دو ربات هم‌زمان یکدیگر را از روبرو دیدند، در یک حلقه تکرار طولانی به خاطر یکدیگر صبر نکنند و یکی از آنها زودتر راه بیفتد و دیگری را بعنوان مانع ایستا ببیند. از آنجا که محدوده را خلوت در نظر گرفته‌ایم، پس احتمال اینکه یک ربات مدت زمان نامحدود منتظر بماند بسیار ناچیز و تقریباً مساوی صفر است. در واقع، برای رخ دادن انتظار نامحدود اولاً باید دو ربات در حال حرکت به طور هم‌زمان از روبرو به فاصله کمتر از  $D_T$  یکدیگر برسند و حسگرهای آنها وجود مانع



پویا در این فاصله را تشخیص بدهد. ثانیاً باید هر دو ربات به صورت نامحدود اعداد تصادفی یکسان تولید کنند و به همان اندازه منتظر بمانند. از آنجا که هرکدام یک عدد تصادفی بین ۱ تا ۳ تولید می‌کند، احتمال یکسان بودن دو عدد در هر بار تولید عدد تصادفی،  $\frac{1}{3}$  است. بنابراین احتمال این که دو بار اعداد یکسان تولید شود،  $\frac{1}{3}$  و برای سه بار  $\frac{1}{27}$  و برای تعداد دفعات زیاد، حد این احتمال به صفر میل می‌کند. زمانی که یک مانع متحرک در مسیر ربات نباشد، حفره‌های پیش رو را اولویت‌بندی کرده و سپس با سرعت ثابت (سرعت ربات‌ها ممکن است با هم متفاوت باشد) به سمت حفره‌ای که بیشترین اولویت را دارد حرکت می‌کند.

**+ حرکت:** این مرحله زمانی است که عمل پویش و تصمیم‌گیری انجام شده و ربات به عملگرهای حرکتی ربات دستور چرخش و حرکت را می‌دهد. زمانی که ربات در حال حرکت است، اگر در نزدیکی مسیر خود مانع متحرک مشاهده کند، به مرحله پویش می‌رود و عملیات توقف موقت (به مدت تصادفی بین ۱ تا ۳ ثانیه) را اجرا می‌کند؛ در غیر اینصورت، به حرکت ادامه می‌دهد تا به مقصد موقت برسد و سپس عملیات پویش را تکرار می‌کند. در این الگوریتم، ربات‌ها به عقب حرکت نمی‌کنند اما توانایی چرخش و بازگشت به مکان قبلی را دارند.

مختصات مقصد بعدی:  $[x'_r, y'_r]$

مختصات فعلی ربات:  $[x_r, y_r]$

مختصات حفره  $h$ :  $[x_h, y_h]$

مختصات مقصد نهایی:  $[x_d, y_d]$

(الف) پویش و تصمیم‌گیری:

۱. اگر  $[x_r, y_r] = [x_d, y_d]$  آنگاه برو به مرحله (ج)

۲. تا زمانی که مانع متحرکی در فاصله کمتر از  $D_r$  است:

(آ) یک عدد تصادفی  $t \in [1, 3]$  انتخاب کن

(ب) به مدت  $t$  ثانیه هیچ کاری نکن

۳. مقدار اولویت پیش‌فرض را صفر قرار بده:  $P \leftarrow 0$

۴. مکان فعلی را به عنوان مقدار پیش‌فرض مکان بعدی قرار بده :

$$[x'_r, y'_r] \leftarrow [x_r, y_r]$$

۵. برای هر حفره قابل رؤیت  $h$  :

(آ) مختصات  $[x_h, y_h]$  را محاسبه کن

(ب) مقدار اولویت  $P_h$  را با استفاده از کنترل‌کننده فازی محاسبه کن

(ج) اگر  $P_h > P$  آنگاه  $P \leftarrow P_h$  و  $[x'_r, y'_r] \leftarrow [x_h, y_h]$

(ب) حرکت:

۱. به سمت  $[x'_r, y'_r]$  حرکت کن

۲. اگر به مانع متحرک نزدیک شدی برو به مرحله (الف)

۳. اگر به  $[x'_r, y'_r]$  رسیدی برو به مرحله (الف)

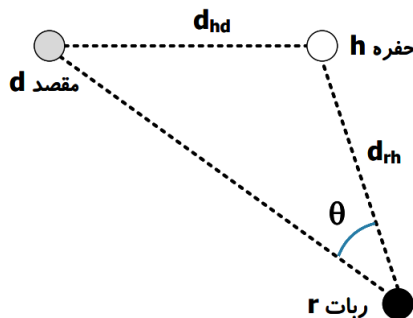
(ج) پایان.

### ۲.۳ کنترل‌کننده فازی

در الگوریتم ۱.۳، در بخش پویس و تصمیم‌گیری، دستور (۵-ب) اولویت حفره  $h$  با استفاده از کنترل‌کننده فازی محاسبه می‌شود. در این بخش، ورودی‌ها، خروجی و ساختار این کنترل‌کننده فازی را شرح می‌دهیم.

شکل ۱ اطلاعات موردنیاز برای محاسبه اولویت یک حفره را نشان می‌دهد. زاویه  $\theta$  میزان انحراف از مسیر مستقیم به سمت مقصد است. یعنی در صورتی که این حفره برای عبور انتخاب شود، ربات برای رسیدن به مقصد خود به میزان  $\theta$  درجه انحراف خواهد داشت. بنابراین هرچه این زاویه بزرگتر باشد، مسیر دارای انحراف بیشتری نسبت به

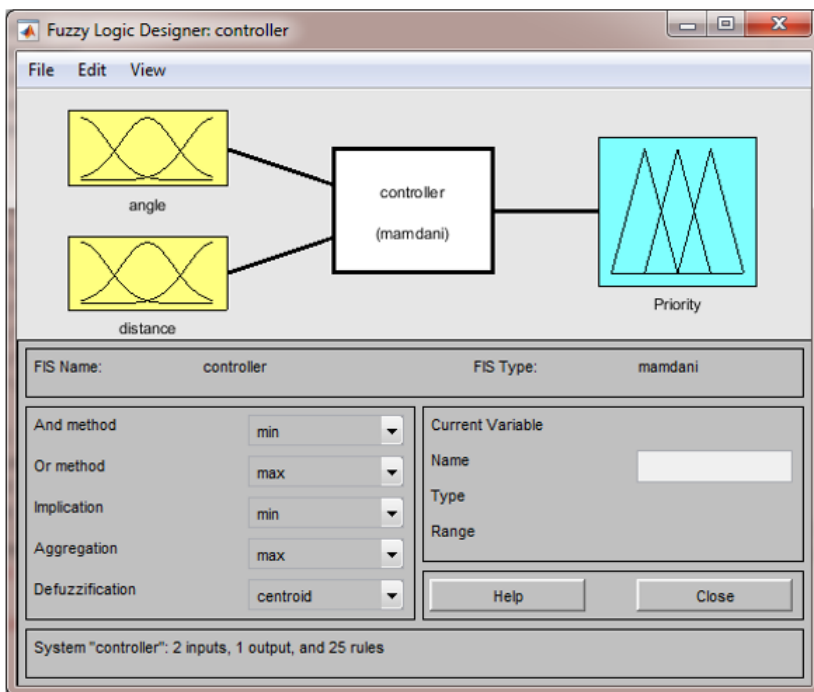
راستای مستقیم با مقصد خواهد بود و معمولاً باعث می‌شود طول مسیر بیشتر شود. البته گاهی ممکن است که یک انحراف نسبتاً زیاد با مسافت کم منجر به پیدا شدن مسیر بهتر شود زیرا نقشه کامل مسیر در دسترس نیست و مشخص نیست که پس از یک جابجایی، چه مسیرهای جدیدی آشکار می‌شوند. همچنین  $d_{rh}$  فاصله مستقیم بین ربات تا حفره و  $d_{hd}$  فاصله مستقیم بین حفره تا مقصد است. بنابراین اگر مابین حفره تا مقصد هیچ مانعی نباشد، ربات از طریق این حفره می‌تواند با پیمودن مسافت  $D = d_{rh} + d_{hd}$  از مکان فعلی خود به مقصد برسد.



شکل ۱: وضعیت نسبی یک ربات، یک حفره و مقصد ربات

کنترل‌کننده فازی از دو پارامتر زاویه  $\theta$  و مسافت  $D$  برای محاسبه اولویت حفره استفاده می‌کند. در ادامه این بخش، مقادیر زبانی متناظر با این دو پارامتر، توابع عضویت آنها و همچنین قواعد ”اگر-آنگاه“ فازی برای استنتاج مقدار اولویت  $P$  و همچنین روش غیرفازی سازی مورد استفاده را بیان می‌کنیم. شکل ۲ شمای کلی کنترل‌کننده فازی طراحی شده را نمایش می‌دهد که مقدار خروجی (*Priority*) با استفاده از ۲۵ قاعده استنتاج و با منطق فازی ممدانی به دست می‌آید.

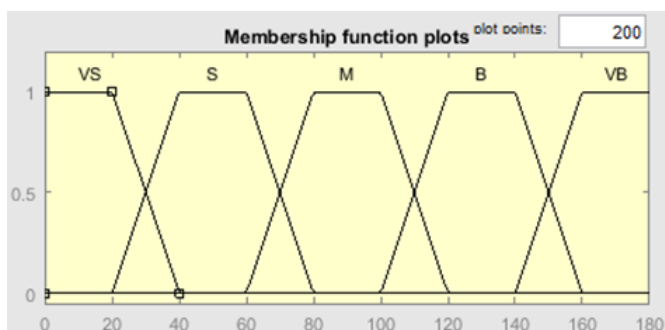
مقدار زاویه  $\theta$  در بازه  $[0, 180]$  قرار دارد زیرا اگر خط واصل ربات به مقصد را به عنوان محور در نظر بگیریم، در حالت کلی انحراف به سمت چپ یا راست این محور را می‌توان با همین بازه و در نظر گرفتن جهت چرخش مدل کرد. مقادیر زبانی مربوط به زاویه  $\theta$  عبارتند از: خیلی تند ( $VS$ )، تند ( $S$ )، متوسط ( $M$ )، باز ( $B$ ) و خیلی باز ( $VB$ ). شکل ۳ توابع عضویت مربوط به این پنج مقدار را نمایش می‌دهد که همه آنها اعداد دوزنقه‌ای هستند و حداکثر یک محدوده ۶۰ درجه را پوشش می‌دهند که در یک سوم



شکل ۲: شمای کلی کنترل‌کننده فازی پیشنهادی

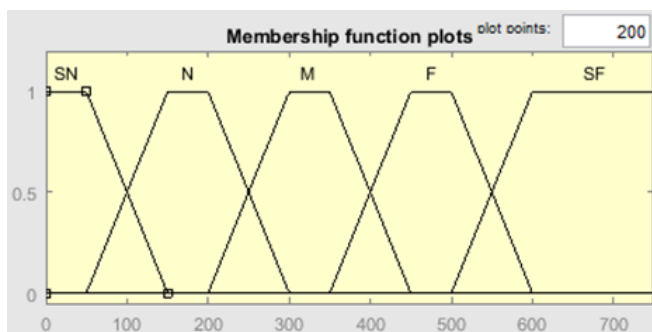
آن درجه عضویت، یک است.

مقدار مسافت  $D$  در بازه  $[0, L_e + W_e]$  قرار دارد؛ زیرا دورترین فاصله ممکن بین یک ربات و مقصد آن به اندازه قطر محدوده است؛ بنابراین مقدار  $D$  وقتی بیشینه می‌شود که ربات و مقصد آن روی گوشه‌های قطری محدوده باشند و ضمناً حفره موردنظر در گوشه‌ای دیگر از محدوده باشد. در این صورت ربات برای عبور از حفره و سپس رسیدن به مقصد، باید مسافتی به اندازه مجموع طول و عرض محدوده را طی کند. مقادیر زبانی مربوط به مسافت  $D$  عبارتند از: خیلی نزدیک ( $V.S$ )، نزدیک ( $S$ )، متوسط ( $M$ )، دور ( $B$ ) و خیلی دور ( $V.B$ ). شکل ۴ توابع عضویت مربوط به این مقادیر زبانی را برای بازه حقیقی  $[0, 750]$  نمایش می‌دهد. البته بازه اصلی تعریف شده برای محیط آزمایش که مربعی به ضلع ۵۰۰ متر است، بازه  $[0, 1000]$  است اما توابع عضویت مقادیر زبانی دقیقاً همین پنج نمودار ذوزنقه‌ای هستند. هر نمودار عضویت (به جز  $SF$ ) در یک بازه به طول حداکثر ۳۰۰ متر دارای مقدار بزرگتر از صفر است که در پنجاه متر از این بازه، مقدار عضویت یک است. در تابع عضویت  $SF$  (خیلی دور)، فاصله‌های ۶۰۰ متر به بالا دارای



شکل ۳: توابع عضویت مربوط به مقادیر زبانی زاویه  $\theta$

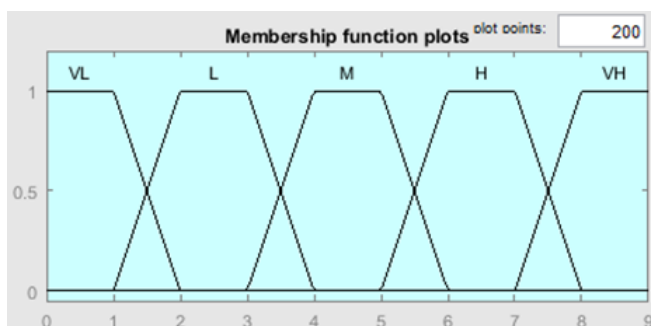
عضویت یک هستند.



شکل ۴: توابع عضویت مربوط به مقادیر زبانی مسافت  $D$

خروجی کنترل‌کننده فازی، اولویت حفره است. مقدار اولویت  $P$  در بازه  $[0, 9]$  قرار دارد به طوری که ۹ بالاترین و ۰ پایین‌ترین اولویت است. پس از آن‌که ربات در مرحله پوشش و تصمیم‌گیری مقدار اولویت تمام حفره‌های قابل رؤیت را محاسبه کرد، به سمت حفره‌ای حرکت می‌کند که بیشترین اولویت را دارد. قواعد استنتاج فازی، از مقادیر زبانی متناظر با اولویت پشتیبانی می‌کنند و پس از استنتاج، با استفاده از عمل غیرفازی سازی، یکی از مقادیر صحیح ۰ تا ۹ به عنوان خروجی به دست می‌آید. مقادیر زبانی مربوط به اولویت  $P$  عبارتند از: خیلی بالا ( $VH$ )، بالا ( $H$ )، متوسط ( $M$ )، پایین ( $L$ ) و خیلی پایین ( $VL$ ). شکل ۵ توابع عضویت مربوط به این پنج مقدار زبانی را نمایش می‌دهد که همه آنها اعداد دوزنقه‌ای هستند. هر نمودار عضویت یک بازه حداکثر به طول ۳ را پوشش می‌دهد (دارای مقدار بزرگتر از صفر است) و در یک سوم این بازه مقدار

عضویت، یک است.



شکل ۵: توابع عضویت مربوط به مقادیر زبانی اولویت  $P$

قواعد استنتاج تعریف شده در جدول ۱ با هدف هدایت ربات در مسیر تقریباً مستقیم و جلوگیری از انحراف زیاد طراحی شده است. البته این قواعد به گونه‌ای طراحی شده که انحراف زیاد در مسیر را برای فاصله‌های کوتاه (حفره‌های نزدیک به ربات) می‌پذیرد؛ زیرا اگر ربات فاصله کوتاهی را از مسیر اصلی کاملاً منحرف شود، افزایش قابل توجهی در طول مسیر کلی ایجاد نمی‌شود. به طور کلی، این قواعد بر مبنای این راهبرد شکل گرفته‌اند که اهمیت منحرف نشدن از مسیر بسیار زیاد است اما در فاصله‌های خیلی نزدیک این اهمیت تا حدودی قابل چشم‌پوشی است. به عنوان مثال، ستون  $V/S$  شامل سه اولویت "خیلی بالا" و دو اولویت "بالا" است اما در ستون‌های بعدی که مربوط بزرگتر شدن زاویه انحراف است، مقدار اولویت‌ها کاهش می‌یابد. همین وضعیت برای سطرها اتفاق افتاده است؛ یعنی در هر ستون، هرچه مسافت کمتر باشد اولویت بیشتر است و با افزایش مسافت تخمینی برای عبور از حفره مورد نظر، اولویت آن حفره کاهش می‌یابد. همچنین، این کنترل‌کننده از روش غیرفازی سازی مبتنی بر محاسبه مرکز ثقل استفاده می‌کند.

## ۴ نتایج شبیه‌سازی

در این بخش، ابتدا آزمایش‌های انجام شده برای تنظیم پارامترها و استراتژی اجتناب از موانع پویا را بیان می‌کنیم و سپس نتایج شبیه‌سازی الگوریتم پیشنهادی و مقایسه آن با الگوریتم [۱۹] را مورد بررسی قرار می‌دهیم.

جدول ۱: لیست قواعد استنتاج کنترل‌کننده فازی پیشنهادی

زاویه $\theta$					مسافت $D$	
$VB$	$B$	$M$	$S$	$VS$		
L	M	H	VH	VH		$SN$
L	L	M	H	VH		$N$
VL	L	M	H	VH		$M$
VL	VL	L	H	H		$F$
VL	VL	L	M	H	$SF$	

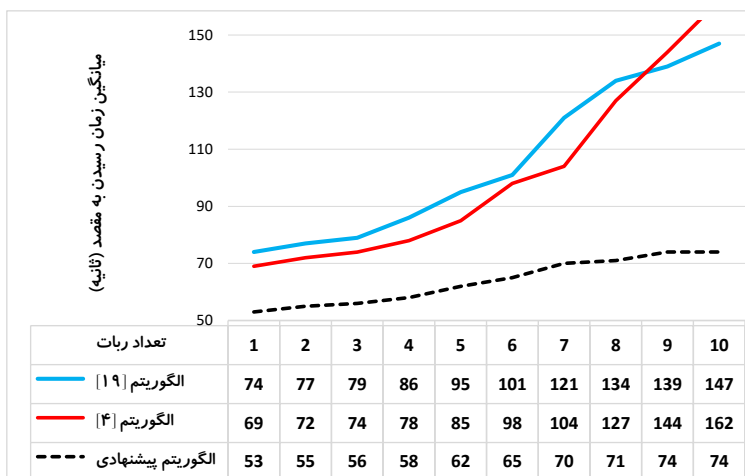
## ۱.۴ شرایط و محیط پیاده‌سازی

الگوریتم پیشنهادی در محیط نرم‌افزار Matlab پیاده‌سازی و کنترل‌کننده فازی آن نیز توسط جعبه ابزار این نرم‌افزار آماده شده است. پارامترهای مسئله مانند محدوده، مبدأ، مقصد و سرعت ربات‌ها و همچنین مختصات موانع توسط یک فایل متنی به‌عنوان ورودی به برنامه داده می‌شود. پس از اجرای کد برنامه، ابتدا مراحل پیش پردازش مانند شبیه‌سازی محیط انجام شده و سپس الگوریتم ۱.۳ اجرا می‌شود. خروجی برنامه، مکان‌های توقف، مدت توقف و مسیری است که هر ربات باید طی کند تا به مقصد برسد.

محدوده آزمایش یک مربع به ضلع ۵۰۰ متر در نظر گرفته شده و تعدادی مانع ایستا به صورت تصادفی در آن قرار داده شده است. هر مانع ایستا یک چندضلعی محدب غیرشفاف و غیرقابل نفوذ است که در طول اجرای الگوریتم، مختصات رئوس آن ثابت می‌ماند. محیط از نظر موانع پویا خلوت است اما از نظر موانع ایستا لزوماً خلوت نیست. برای هر اجرای الگوریتم، تعداد مشخصی ربات با مبدأ و مقصد تصادفی و همچنین سرعت تصادفی بین ۱ تا ۵ متر بر ثانیه تولید شده است. این مشخصات محیط برای اجرای الگوریتم پیشنهادی و همچنین الگوریتم‌های [۱۹] و [۴] یکسان در نظر گرفته شده است. تعداد ربات‌ها در هر آزمایش از ۱ تا ۱۰ دستگاه در نظر گرفته شده تا محیط از جنبه تعداد موانع پویا خلوت باشد و برای هر تعداد ربات، آزمایش ده بار تکرار شده و سپس میانگین مجموع مسافت‌های طی شده و میانگین مجموع زمان‌های طی شده توسط ربات‌ها محاسبه و مقایسه شده است.

## ۲.۴ نتایج تجربی پیاده‌سازی

شکل ۶ نمودار و جدول میانگین زمان رسیدن ربات‌ها به مقصد را نمایش می‌دهد. در اینجا نمودار آبی مربوط به الگوریتم [۱۹]، نمودار قرمز مربوط به الگوریتم [۴] و نمودار خط‌چین سیاه مربوط به الگوریتم پیشنهادی است. روند صعودی هر سه نمودار نشان می‌دهد که با افزایش تعداد ربات‌ها در محیط، زمان رسیدن هر ربات به مقصد (به صورت میانگین) افزایش داشته است؛ زیرا تعداد موانع پویا در محیط بیشتر شده و در نتیجه، در الگوریتم‌های [۱۹] و [۴] هر ربات تعداد دفعات بیشتری از مسیر خود منحرف می‌شود و در الگوریتم پیشنهادی نیز هر ربات باید زمان‌های توقف بیشتری داشته باشد. اطلاعات نمودارها نشان می‌دهد که الگوریتم پیشنهادی کارایی بالاتری نسبت به الگوریتم‌های [۱۹] و [۴] دارد؛ زیرا در تمام آزمایش‌ها دارای کمترین میانگین زمانی رسیدن ربات به مقصد است. همچنین روند رشد (شیب نمودار) در الگوریتم پیشنهادی کمتر است. از سوی دیگر، زمانی که فقط یک ربات در محیط وجود دارد، الگوریتم پیشنهادی بهتر عمل کرده و نشان می‌دهد که برای محیط‌های ایستا نیز کارایی بهتری دارد.

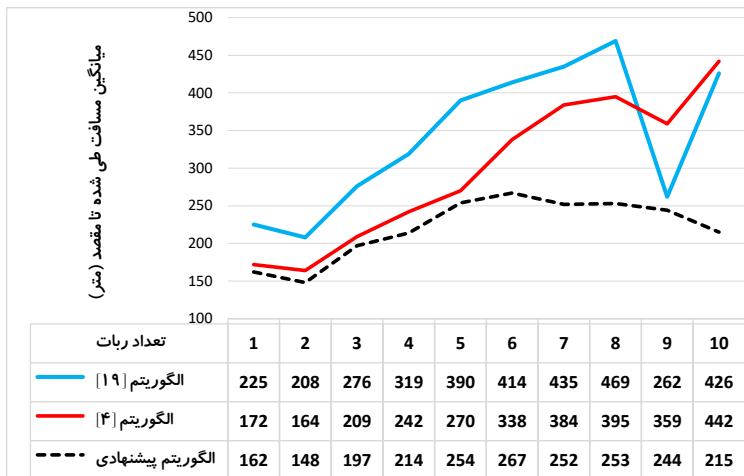


شکل ۶: میانگین زمان رسیدن ربات‌ها به مقصد

شکل ۷ نمودار و جدول میانگین مسافت طی شده توسط ربات‌ها برای رسیدن به مقصد را نمایش می‌دهد. در اینجا نمودار آبی مربوط به الگوریتم [۱۹]، نمودار قرمز مربوط به الگوریتم [۴] و نمودار خط‌چین سیاه مربوط به الگوریتم پیشنهادی است. با



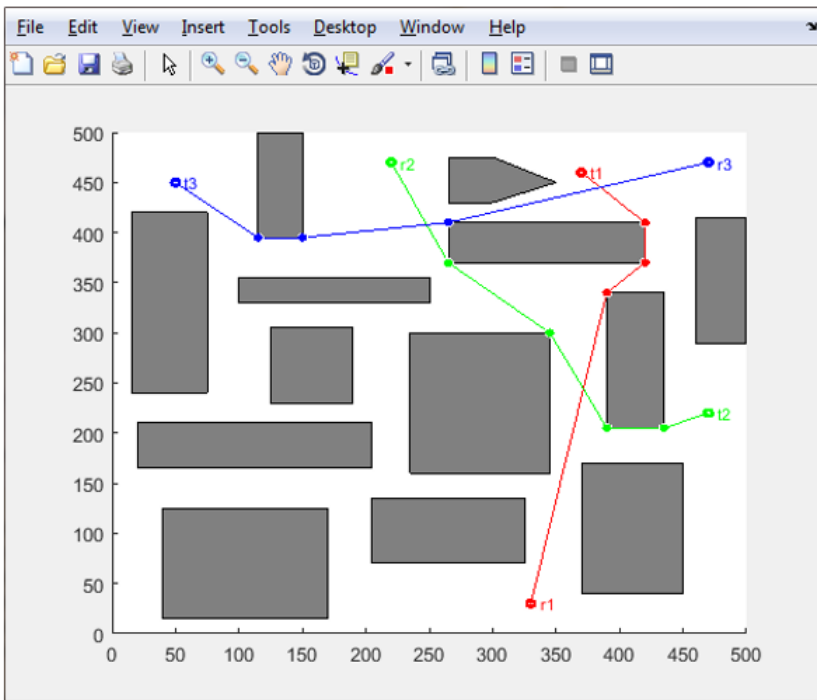
افزایش تعداد ربات‌ها در محیط، در الگوریتم‌های [۱۹] و [۴] هر ربات تعداد دفعات بیشتری از مسیر خود منحرف می‌شود و معمولاً میانگین مسافت طی شده بیشتر می‌شود اما در الگوریتم پیشنهادی، در هر رویارویی دو ربات، یکی از آنها توقف می‌کند و از مسیر خود منحرف نمی‌شود و دیگری هم در صورت روبرو شدن از جلو، مقداری نسبت به مسیر منحرف می‌شود؛ بنابراین میانگین مسافت طی شده رشد کمتری دارد. اطلاعات نمودار نشان می‌دهد که الگوریتم پیشنهادی در مقایسه با الگوریتم‌های [۱۹] و [۴] کارایی بالاتری دارد؛ زیرا در تمام آزمایش‌ها میانگین مسافت طی شده توسط هر ربات کمتر از آنها است. همچنین روند رشد نمودار در الگوریتم پیشنهادی کمتر است.



شکل ۷: میانگین مسافت طی شده توسط ربات‌ها تا مقصد

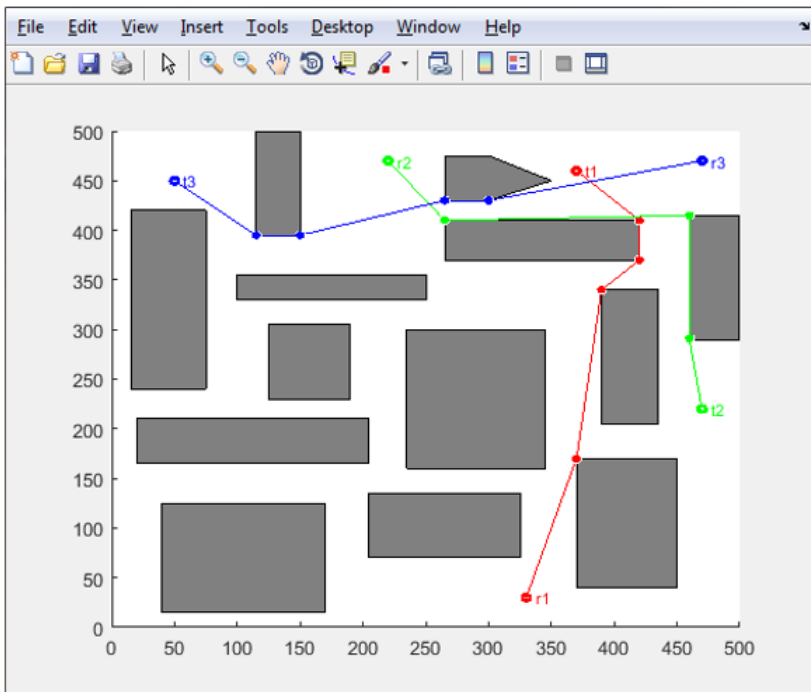
در شکل ۸ خروجی نهایی به دست آمده پس از اجرای الگوریتم پیشنهادی در محیط شامل سه ربات  $r_1$ ،  $r_2$  و  $r_3$  با مقاصد  $t_1$ ،  $t_2$  و  $t_3$  مشاهده می‌شود. در اینجا سرعت زمان شروع به حرکت ربات‌ها به گونه‌ای در نظر گرفته شده با هم در مسیر حرکت تداخل داشته باشند. راهبرد توقف تا زمان عبور مانع پویا موجب شده تا الگوریتم پیشنهادی مسیرهای خوبی برای آنها پیدا کند؛ زیرا در بیشتر موارد، مسیر نهایی هر ربات شبیه حالتی می‌شود که موانع پویا در محیط حضور ندارند. البته در برخی موارد ممکن است توقف و شروع مجدد به حرکت، موجب انتخاب حفره جدید برای عبور شود.

در شکل ۹ خروجی نهایی اجرای الگوریتم پیشنهادی با راهبرد تغییر جهت و عدم



شکل ۸: مسیر نهایی ربات‌ها در اجرای الگوریتم پیشنهادی با راهبرد امکان توقف در شرایط تداخل حرکت در یک محیط شامل سه ربات

امکان توقف در همان محیط مشاهده می‌شود. در واقع، تمام شرایط مانند آزمایش قبل است و فقط راهبرد مسیریابی تغییر کرده است. در اینجا، عدم امکان توقف موجب اجبار در تغییر مسیر ربات در برخی موارد شده و ربات مسیر طولانی‌تری را پیموده است. به عنوان مثال، ربات  $r_2$  که با رنگ سبز نمایش داده شده، در اوایل مسیر خود با ربات  $r_3$  روبرو شده و تغییر جهت داده است. این تغییر جهت، مسیر پیموده شده تا مقصد را به کلی عوض کرده و باعث افزایش طول مسیر پیموده شده توسط ربات  $r_2$  شده است. همچنین تغییر جهت جزئی ربات  $r_3$  در نتیجه رویارویی با ربات  $r_1$  نیز قابل توجه است. هرچند این تغییر جهت موجب تغییر محسوس در مسیر نهایی ربات  $r_3$  نشده اما باعث شده تا ربات  $r_3$  بر ربات  $r_1$  تأثیر بگذارد و مسیر آن را تغییر دهد.



شکل ۹: مسیر نهایی ربات‌ها در اجرای الگوریتم پیشنهادی با راهبرد امکان تغییر جهت در شرایط تداخل حرکت در یک محیط شامل سه ربات

## ۵ نتیجه‌گیری

### ۱.۵ خلاصه پژوهش و نتایج به دست آمده

در این مقاله، یک الگوریتم مسیریابی برای ربات‌های متحرک در محیط‌های پویای ناشناخته خلوت پیشنهاد شده است که با استفاده از منطق فازی، ربات را در هرگام به سمت مقصد هدایت می‌کند. الگوریتم پیشنهادی از دو بخش پویا و تصمیم‌گیری و حرکت تشکیل شده و برای جلوگیری از برخورد ربات‌های متحرک استراتژی توقف با زمان تصادفی را دارد. همچنین یک کنترل‌کننده فازی، حفره‌های قابل رؤیت مابین موانع را برای عبور اولویت‌بندی می‌کند. نتایج شبیه‌سازی، کارایی الگوریتم پیشنهادی برای کمینه‌سازی زمان و مسافت حرکت ربات را تأیید می‌کند و نشان می‌دهد که برای محیط‌هایی که تردد موانع پویا در آنها کم است، توقف‌های کوتاه بهتر از تغییر مسیر است و این رویکرد موجب کاهش میزان مسافت نهایی طی شده تا مقصد می‌شود.

## ۲.۵ پیشنهاد برای پژوهش‌های آتی

کنترل‌کننده فازی مورد استفاده در این مقاله تطبیق‌پذیر نیست؛ زیرا توابع عضویت قابلیت اصلاح در حین عملیات را ندارد. می‌توان اصلاح این توابع عضویت با استفاده از روش‌های ابتکاری یا فراابتکاری را به عنوان یک زمینه پژوهشی دنبال نمود.

یکی از فرضیات ضمنی در این پژوهش، عدم به وجود آمدن بن‌بست در نتیجه تقابل دو ربات در یک حفره یا دالان مابین موانع است. اگر این فرض در نظر گرفته نشود، باید راهکاری برای دور زدن و بازگشت ربات از یک حفره طراحی شود. در [۱۳] یک راهکار خروج از بن‌بست با دور زدن ارائه شده است اما مربوط به تقابل دو ربات در یک دالان نیست و توانایی تصمیم‌گیری در مورد این که کدام ربات به مسیر خود ادامه دهد و کدام یک دور بزند را ندارد. بنابراین، راهکار خروج از بن‌بست‌های پویا یک مسئله جداگانه و قابل پژوهش خواهد بود.

در این پژوهش، موانع پویا ربات‌ها هستند. بنابراین در صورت ایست یک ربات، ربات دیگر نیز ایست می‌کند یا تغییر مسیر می‌دهد و برخورد اتفاق نمی‌افتد. برای حالتی که موانع پویا هوشمند نباشند و احتمال برخورد با آنها در حالت ایست ربات نیز وجود دارد، راهکار اجتناب یک صورت مسئله مجزا است و می‌تواند به عنوان زمینه پژوهشی دنبال شود.

## مراجع

- [1] Abed, M.S., Lutfy, O.F. and Al-Doori, Q. (2021) A Review on Path Planning Algorithms for Mobile Robots. *Engineering and Technology Journal*, 39(5A), 804-820.
- [2] Caraveo, C., Valdez, F. and Castillo, O. (2016) Optimization of fuzzy controller design using a new bee colony algorithm with fuzzy dynamic parameter adaptation. *Applied Soft Computing*, 43, 131-142.

- [3] Chen, Y., Liang, J., Wang, Y., Pan, Q., Tan, J. and Mao, J. (2020) Autonomous mobile robot path planning in unknown dynamic environments using neural dynamics. *Soft Computing*, 24(18), 13979-13995.
- [4] Elmi, Z. and Efe, M. Ö. (2021) Online path planning of mobile robot using grasshopper algorithm in a dynamic and unknown environment. *Journal of Experimental & Theoretical Artificial Intelligence*, 33(3), 467-485.
- [5] Ge, S.S. and Cui, Y.J. (2002) Dynamic motion planning for mobile robots using potential field method. *Autonomous robots*, 13(3), 207-222.
- [6] Juang, C.F., Lin, C.H. and Bui, T.B. (2018) Multiobjective rule-based cooperative continuous ant colony optimized fuzzy systems with a robot control application. *IEEE transactions on cybernetics*, 50(2), 650-663.
- [7] Lin, X., Wang, C., Wang, K., Li, M. and Yu, X. (2021) Trajectory planning for unmanned aerial vehicles in complicated urban environments: A control network approach. *Transportation Research Part C: Emerging Technologies*, 128, 103120.
- [8] Martínez, E.A.R., Caron, G., Pégard, C. and Alabazares, D.L. (2020) Photometric path planning for vision-based navigation. In 2020 IEEE International Conference on Robotics and Automation (ICRA), 9007-9013, IEEE.
- [9] Mitchell, J.P., Bruer, G., Dean, M.E., Plank, J.S., Rose, G.S. and Schuman, C.D. (2017) NeoN: Neuromorphic control for autonomous robotic navigation. In 2017 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS), 136-142, IEEE.
- [10] Mohanty, P.K. and Dewang, H.S. (2021) A smart path planner for wheeled mobile robots using adaptive particle swarm optimization. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 43(2), 1-18.

- [11] Rahmani, W. and Rakhmania, A.E. (2022) Mobile Robot Path Planning in a Trajectory with Multiple Obstacles Using Genetic Algorithms. *Journal of Robotics and Control (JRC)*, 3(1), 1-7.
- [12] Rusu, P., Petriu, E.M., Whalen, T.E., Cornell, A. and Spoelder, H.J. (2003) Behavior-based neuro-fuzzy controller for mobile robot navigation. *IEEE transactions on instrumentation and measurement*, 52(4), 1335-1340.
- [13] Sharma, K.D., Chatterjee, A. and Rakshit, A. (2018) Experimental study II: Vision-based navigation of mobile robots. In *Intelligent Control*, 243-280, Springer, Singapore.
- [14] Tran, V.P., Garratt, M.A. and Petersen, I.R. (2021) Multi-vehicle formation control and obstacle avoidance using negative-imaginary systems theory. *IFAC Journal of Systems and Control*, 15, 100117.
- [15] Uriol, R. and Moran, A. (2017) April. Mobile robot path planning in complex environments using ant colony optimization algorithm. In *2017 3rd international conference on control, automation and robotics (ICCAR)*, 15-21, IEEE.
- [16] Van Den Berg, J., Stilman, M., Kuffner, J., Lin, M. and Manocha, D. (2009) Path planning among movable obstacles: a probabilistically complete approach. In *Algorithmic Foundation of Robotics VIII*, 599-614, Springer, Berlin, Heidelberg.
- [17] Wu, Z., Chen, Y., Liang, J., He, B. and Wang, Y. (2021) ST-FMT\*: A Fast Optimal Global Motion Planning for Mobile Robot. *IEEE Transactions on Industrial Electronics*.
- [18] Zhang, Z., Zheng, L., Chen, Z., Kong, L. and Karimi, H.R. (2020) Mutual-collision-avoidance scheme synthesized by neural networks for dual redundant robot manipulators executing cooperative tasks. *IEEE transactions on neural networks and learning systems*, 32(3), 1052-1066.

- [19] Zhao, R. and Lee, H.K. (2017) Fuzzy-based path planning for multiple mobile robots in unknown dynamic environment. *Journal of Electrical Engineering and Technology*, 12(2), 918-925.
- [20] Zhou, Y., Hu, H., Liu, Y., Lin, S.W. and Ding, Z. (2020) A distributed method to avoid higher-order deadlocks in multi-robot systems. *Automatica*, 112, 108706.