
بسم الله
الرحمن الرحیم

دانشکده علوم ریاضی

رشته ریاضی کاربردی، گرایش تحقیق در عملیات

پایان‌نامه کارشناسی ارشد

مسیریابی مبتنی بر منطق فازی برای ربات‌های متحرک در محیط‌های پویای ناشناخته

نگارنده: زهرا براتی

استادان راهنما

دکتر جعغر فتحعلی
دکتر سمیه مغاری

بهمن ۱۳۹۷

تقدیم به اولین معلمان زندگیم
پدر و مادر عزیزم.

اینك که به فضل خداوند این پروژه به سرانجام رسیده، وظیفه‌ی خود می‌دانم از زحمات بی‌دریغ استاد گرامی جناب آقای دکتر جعفر فتحعلی و خانم دکتر سمیه مغاری که در اجرای این پروژه اینجانب را یاری فرموده‌اند، کمال تشکر و قدردانی را داشته باشم.
همواره از خداوند متعال سلامتی و توفیق روز افزون شما را خواهانم.

زهرا براتی
بهمن ۱۳۹۷

تعهد نامه

اینجانب **زهرا براتی** دانشجوی کارشناسی ارشد رشته **ریاضی کاربردی علوم ریاضی** دانشگاه شاهرود، نویسنده پایان‌نامه با عنوان **مسیریابی مبتنی بر منطق فازی برای ربات‌های متحرک در محیط‌های پویای ناشناخته** ، تحت راهنمایی **جعفر فتحعلی** و **سمیه مغاری** متعهد می‌شوم:

- تحقیقات در این پایان‌نامه توسط اینجانب انجام شده است و از صحت و اصالت برخوردار است.
- در استفاده از نتایج پژوهش‌های دیگر پژوهش‌گران، به مرجع مورد استفاده استناد شده است.
- مطالب این پایان‌نامه، تا کنون توسط خود، یا فرد دیگری برای دریافت هیچ نوع مدرک یا امتیازی در هیچ‌جا ارایه نشده است.
- حقوق معنوی این اثر، به دانشگاه صنعتی شاهرود تعلق دارد، و مقالات مستخرج با نام " دانشگاه صنعتی شاهرود " یا " Shahrood University of Technology " به چاپ خواهد رسید.
- حقوق معنوی تمام افرادی که در به‌دست آوردن نتایج اصلی پایان‌نامه تاثیرگذار بوده‌اند، در مقالات مستخرج از پایان‌نامه رعایت می‌گردد.
- در تمام مراحل انجام این پایان‌نامه، در مواردی که از موجود زنده (یا بافت‌های آنها) استفاده شده است، ضوابط و اصول اخلاقی رعایت شده است.
- در تمام مراحل انجام این پایان‌نامه، در مواردی که به حوزه اطلاعات شخصی افراد دسترسی یافته (یا استفاده شده است)، اصل رازداری و اصول اخلاق انسانی رعایت شده است.

زهرا براتی
بهمن ۱۳۹۷

مالکیت نتایج و حق نشر

- تمام حقوق معنوی این اثر و محصولات آن (مقالات مستخرج، کتاب، برنامه‌های رایانه‌ای، نرم‌افزارها و تجهیزات ساخته شده) متعلق به دانشگاه صنعتی شاهرود می‌باشد. این مطلب باید به نحو مقتضی، در تولیدات علمی مربوطه ذکر شود.
- استفاده از اطلاعات و نتایج موجود در این پایان‌نامه بدون ذکر منبع مجاز نمی‌باشد.

چکیده

در این پایان نامه به مطالعه‌ی نحوه‌ی طراحی کنترل کننده‌های فازی برای برنامه ریزی مسیر ربات‌ها در محیط‌های ناشناخته پویا می‌پردازیم. ناشناخته بودن محیط به معنای در دسترس نبودن نقشه کل مسیرها و موانع موجود است و اگر برخی موانع متحرک باشند، محیط را پویا می‌نامیم. هدف اصلی ما برنامه ریزی مسیر حرکت گروهی از ربات‌ها از حالت دینامیکی اولیه به حالت نهایی است در حالی که از برخورد با یکدیگر ، موانع پویا و ایستا اجتناب کنند. طبق فرض مسئله، هیچ یک از ربات‌ها اطلاعات کاملی از محیط ندارند و دانش خود را با یکدیگر به اشتراک نمی‌گذارند. بنابراین یافتن مسیر بهینه همیشه امکان‌پذیر نیست و یافتن مسیر خوب (نزدیک به بهینه) بسیار ارزشمند است. در این پایان‌نامه، ابتدا مسئله را به صورت برنامه ریزی خطی صحیح بازنویسی می‌کنیم. سپس با استفاده از کنترل‌کننده فازی به عنوان یک روش فراابتکاری نحوه مسیریابی یک ربات در محیط ناشناخته ایستا را بررسی می‌کنیم. در ادامه، حالت قبل را گسترش داده و به تشریح مسیریابی همزمان چند ربات در محیط ناشناخته پویا با استفاده از کنترل‌کننده‌فازی می‌پردازیم. در پایان، الگوریتم پیشنهادی برای بهینه‌سازی چندهدفه مسیریابی ربات‌ها در محیط‌های پویای خلوت را ارائه نموده و نشان خواهیم داد این رویکرد قابلیت یافتن پاسخ نزدیک به بهینه را در زمان بسیار کوتاه دارد.

کلمات کلیدی: برنامه‌ریزی مسیر چندرباتی، محیط ناشناخته پویا، کنترل کننده فازی، بهینه‌سازی چندهدفه.

فهرست مطالب

فهرست تصاویر

فهرست جداول

فصل ۱

۱.۱ مقدمه

فناوری ربات خودکار به‌عنوان بخش مهمی از موضوع رباتیک، در سال‌های اخیر مورد استقبال گسترده‌تری قرارگرفته است. ربات‌ها کاربردهای زیادی در بخش‌هایی مانند کاوش در فضا، دستگاه‌های تنظیم‌کننده کارخانه‌ها، استخراج معدن، موارد پزشکی، نظامی و خدمات دارند که می‌توانند در نیروی کار صرفه‌جویی کنند. در این کاربردها مسئله تعیین مسیر ربات‌ها یکی از موضوعات چالش‌برانگیز و پرطرفدار است.

۲.۱ مفاهیم پایه ریاضیات فازی برای کنترل فازی و مدل‌سازی

کنترل فازی و مدل‌سازی فقط یک بخش کوچک از ریاضیات فازی را استفاده می‌کنند که این قسمت ازلحاظ ریاضی بسیار ساده و دارای مفهوم آسانی است. در این فصل برخی از مفاهیم اساسی پیرامون مجموعه‌های فازی و منطق فازی را معرفی می‌کنیم. موارد تحت پوشش در این فصل به‌عنوان یک پایه مقدماتی برای خواننده به‌منظور درک کنترل فازی و مدل‌سازی در این پایان‌نامه ارائه شده است. تعاریف این فصل از [۱] برداشت شده است .

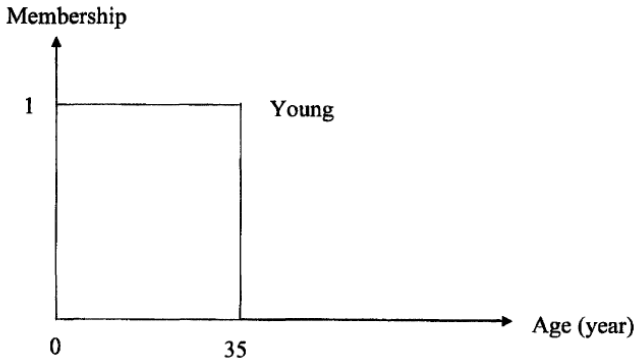
۳.۱ مجموعه‌های کلاسیک، مجموعه‌های فازی و منطق فازی

۱.۳.۱ تحدیدی از مجموعه‌های کلاسیک

در نظریه‌ی مجموعه‌های کلاسیک، عضویت یک شی متعلق به یک مجموعه، تنها می‌تواند یکی از دو مقدار صفر یا یک باشد. یک شی یا به‌طور کامل متعلق به یک مجموعه است یا به آن تعلق ندارد. مجموعه‌های کلاسیک به‌خوبی از مفهوم سیاه و سفید برخوردارند مانند «صندلی‌ها، کشتی‌ها و درختان»؛ که ابهام کمی در آن وجود دارد. با این حال، این‌ها برای توصیف واقع‌گرایانه مفاهیم مبهم کافی نیستند. در زندگی روزمره ما مفاهیم بسیاری وجود دارد که می‌توانیم به راحتی آن‌ها را توضیح دهیم، اما ریاضیات کلاسیک ازجمله نظریه مجموعه‌ها نمی‌تواند به شیوه‌ای منطقی رفتار کند.

به‌عنوان‌مثال مفهوم «جوان» را در نظر بگیرید. برای هر شخص خاصی سن او دقیق است. با این حال، اختصاص سن خاصی به‌عنوان «جوان» شامل فازی‌سازی است؛ و گاهی اوقات گیج‌کننده و مشکل است که چه سنی جوان است و کدام سن جوان نیست. ماهیت چنین سوالاتی قطعی است و هیچ ارتباطی با مفاهیم تصادفی مانند احتمال یا امکان‌پذیری ندارد.

یک مجموعه کلاسیک فرضی «جوان» در شکل ۱.۱ نشان داده‌شده است. این مجموعه



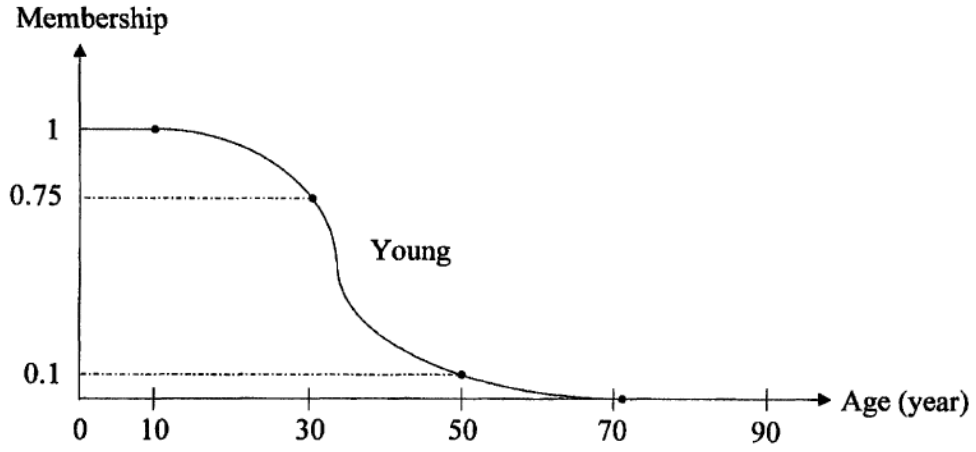
شکل ۱.۱: یک توضیح ممکن از مفهوم مبهم «جوان» به‌وسیله‌ی مجموعه کلاسیک

به دلیل تغییر ناگهانی مقدار عضویت از یک به صفر در مقدار ۳۵ ، غیرمنطقی است. اگرچه مقطع سنی مختلف در نمودار عضویت در حال تغییر از صفر تا یک که در ابتدای مسئله استفاده می‌شود، ممکن است وجود داشته باشد؛ اما چرا فرد ۳۴.۹ ساله به‌طور کامل جوان است درحالی‌که یک فرد ۳۵.۱ ساله اصلا جوان نیست؟ هیچ مجموعه کلاسیکی نمی‌تواند به‌طور واقع‌گرایی ازلحاظ کمی یا کیفی، مفهوم کلمه‌ی مبهم جوان را به‌طور منطقی با آنچه انسان از مفهوم جوان می‌پندارد ، بیان کند. این مثال ساده به معنای بی‌اعتبار کردن

نظریه مجموعه‌های کلاسیک نیست، بلکه قصد دارد نشان دهد که مجموعه‌های کلاسیک و مجموعه‌های فازی دو ابزار مختلف و مکمل هستند که هرکدام دارای نقاط قوت، محدودیت و دامنه‌های کاربردی هستند.

۲.۳.۱ مجموعه‌های فازی

نظریه‌ی مجموعه‌های فازی توسط پروفسور لطفی عسکرزاده در دانشگاه کالیفرنیا در سال ۱۹۶۵ پیشنهاد شده است. این نظریه بر پایه محاسباتی با کلمات استوار است. نظریه مجموعه‌های فازی مقادیر عضویت صفر و یک را از یک مجموعه کلاسیک به یک تابع عضویت در بازه $[۰,۱]$ از مجموعه‌ی فازی تعمیم می‌دهد. در استفاده از این نظریه، سن مربوط به «جوان» را با مقدار عضویت در محدوده‌ی صفر تا یک مرتبط می‌سازد، صفر به معنای عدم اختصاص و یک به معنای اختصاص کامل است. به‌عنوان‌مثال ممکن است فکر کنید که شخص ۱۰ ساله با مقدار عضویت ۱ ، شخص ۳۰ ساله با مقدار عضویت ۰.۷۵ و شخص ۵۰ ساله با مقدار عضویت ۰.۱ جوان است و... این بدان معنی است که هر فرد، با درجه خاص سن خودش «جوان» است. همان‌طور که در شکل ۲.۱ نشان داده شده است، با ترسیم مقادیر عضویت در مقایسه با سنین یک مجموعه فازی «جوان» ایجاد می‌کنیم. منحنی شکل، تابع عضویت مجموعه فازی «جوان» نامیده می‌شود. تمام سنین ممکن از ۰ تا ۱۳۰ سال جامعه موردبررسی را تشکیل می‌دهند. از این مثال، به‌طور طبیعی یک تعریف از مجموعه‌ی فازی به دست می‌آید.



شکل ۲.۱: یک توضیح ممکن از مفهوم مبهم «جوان» به‌وسیله‌ی مجموعه فازی

تعریف ۱.۳.۱. مجموعه فازی : یک مجموعه فازی شامل یک مجموعه جهانی و تابع عضویت است که هر عنصر در مجموعه‌ی جهانی را به یک مقدار عضویت بین ۰ و ۱ می‌برد.از حروف الفبای بزرگ و تیلد (مانند $\tilde{A}$) برای نشان دادن مجموعه‌ی فازی در این پایان‌نامه استفاده

می‌کنیم. یک عنصر با $x \in X$ نشان داده می‌شود، X یک مجموعه‌ی جهانی است و تابع عضویت مجموعه فازی $\tilde{A}$ با $\mu_{\tilde{A}}(x)$ به ازاي هر $x \in X$ نشان داده می‌شود و به‌صورت زیر تعریف مي‌شود :

$$\mu_{\tilde{A}}(x) : X \longrightarrow [۰, ۱]$$

$$\tilde{A}(x) = \{(x, \mu_{\tilde{A}}(x)) | x \in X\}$$

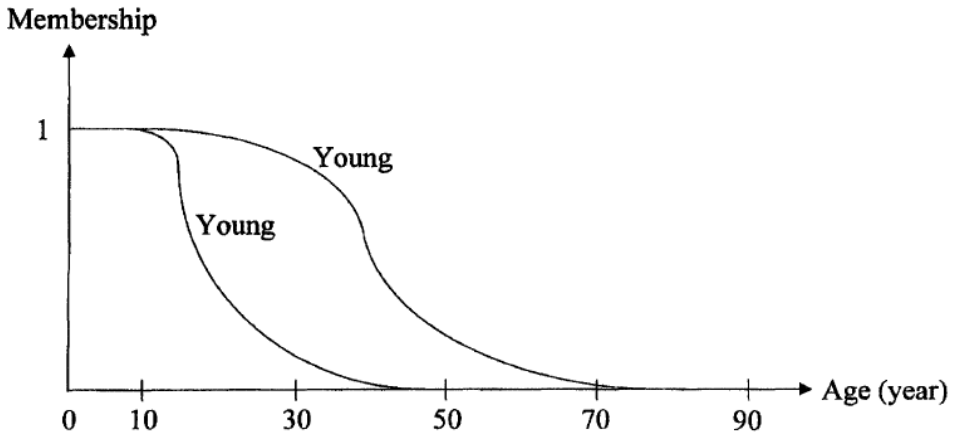
برای مثال سن فوق ، $X = [۰, ۱۳۰]$ است. فرض کنید $\tilde{A}$ به یک مجموعه فازی «جوان» دلالت کند. تابع عضویت آن را با $\mu_{\tilde{A}}(x)$ نشان می‌دهیم که $x \in X$. افراد دارای دیدگاه‌های متفاوتي در مورد مفهوم مبهم هستند. مجموعه‌های فازی می‌توانند به آسانی برای اصلاح کردن این موضوع مورداستفاده قرار گیرند. بعضی از افراد ممکن است فکر کنند سن ۵۰ سالگی با مقدار عضویت ۰.۹ «جوان» است، درحالی‌که ممکن است دیگران سن ۲۰ سالگی با مقدار عضویت ۰.۲ را «جوان» در نظر بگیرند. تابع عضویت‌های مختلفی برای نشان دادن مفهوم «جوان» استفاده می‌شود. شکل ۳.۱ دو تعریف دیگر از مجموعه فازی «جوان» را نشان می‌دهد. نه‌تنها افراد مختلف نسبت به مفهوم مبهم دیدگاه‌هاي متفاوتي دارند ، بلکه تابع عضویت‌هاي مختلفي را نیز براي يك مفهوم مبهم در نظر می‌گیرند . بنابراین یک فرد نیز می‌تواند برای مفهوم «جوان» توابع عضویت مختلفی در نظر بگیرد که با تغییر محدوده سنی در زمینه‌ی ارائه‌شده، تغییر کند. به‌عنوان‌مثال، یک رئیس‌جمهور ۴۰ ساله یک کشور به احتمال زیاد به‌عنوان جوان در نظر گرفته خواهد شد، درحالی‌که یک ورزشکار ۴۰ ساله این‌طور نخواهد بود. دو مجموعه‌ی فازی مختلف «جوان» برای برقراری ارتباط موثر بین این دو موقعیت، مورد نیاز است. این نمونه‌ها نشان می‌دهد که

۱. مجموعه‌های فازی می‌توانند به‌طور کاربردی و کیفی ابهام را در مفاهیم مبهم رفع کنند.

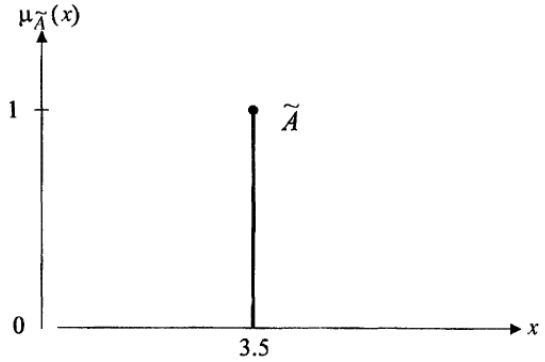
۲. افراد می‌توانند از توابع عضویت مختلف برای توصیف یک مفهوم مبهم یکسان استفاده کنند.

اکنون برخی از تعاریف موردنیاز برای توصیف کنترل‌کننده‌های فازی و مدل‌ها را معرفی می‌کنیم.

تعریف ۲.۳.۱. مجموعه فازی پیوسته :یک مجموعه فازی پیوسته نامیده می‌شود، اگر تابع عضویت آن پیوسته باشد. معمولا کنترل‌کننده‌های فازی و مدل‌ها از مجموعه‌های فازی پیوسته استفاده می‌کنند.



شکل ۳.۱: دو مدل ممکن برای توصیف مفهوم مبهم جوان در مجموعه‌های فازی



شکل ۴.۱: یک مثال برای تابع عضویت مجموعه‌ی فازی یکتا

تعریف ۳.۳.۱. مجموعه‌های فازی یکتا : یک مجموعه فازی که فقط یک عضو از مجموعه‌ی جهانی آن دارای مقدار عضویت غیر صفر و برابر یك باشد را مجموعه فازی یکتا می‌نامیم. در شکل ۴.۱ مجموعه‌ای فازی یکتا نشان داده شده است که در همه‌جای بازه مقدار عضویت صفر دارد به جز در $X = ۳ \cdot ۵$ که مقدار عضویت آن یک است. بیشتر کنترل‌کننده‌ها و مدل‌های فازی، مجموعه‌های فازی یکتا را در نتایج قوانین فازی به کار می‌برند.

تعریف ۴.۳.۱. پشتیبان یک مجموعه فازی: برای یک مجموعه فازی که مجموعه جهانی آن X است، تمام عناصر موجود در X که مقدار عضویت آن‌ها مثبت (غیر صفر) است، پشتیبان مجموعه‌ی فازی گفته می‌شود.

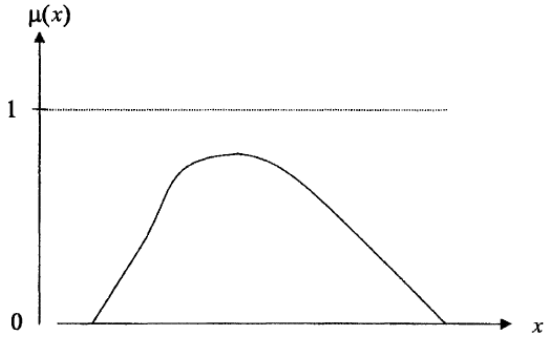
به‌عنوان یک مثال واضح ، پشتیبان مجموعه‌ی فازی «جوان» که در شکل ۲.۱ نشان داده‌شده [۰,۷۰] هست.

تعریف ۵.۳.۱. ارتفاع یک مجموعه فازی: بزرگ‌ترین مقدار عضویت از مجموعه‌ی فازی را ارتفاع مجموعه‌ی فازی می‌نامند.

به‌عنوان‌مثال، ارتفاع مجموعه‌ی فازی «جوان» در شکل ۲.۱ مقدار یک است، ارتفاع مجموعه‌های فازی که در کنترل‌کننده‌های فازی و مدل‌ها استفاده می‌شود معمولا یک است.

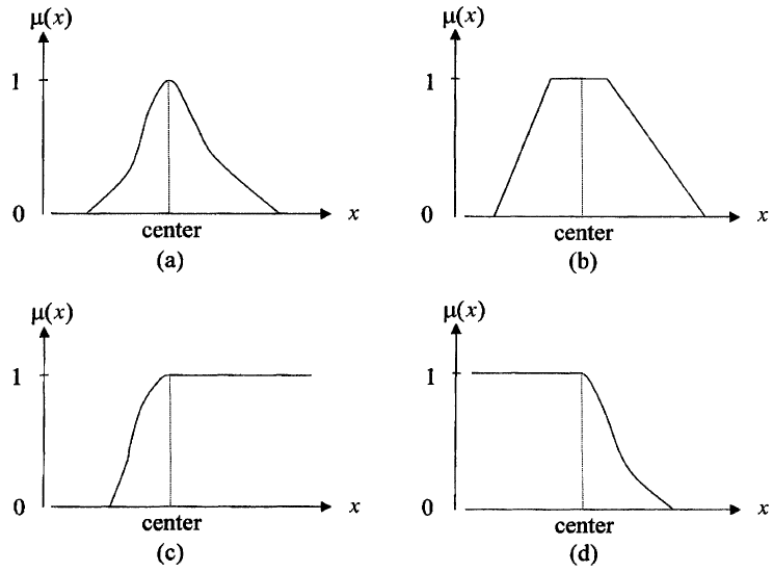
تعریف ۶.۳.۱. مجموعه‌ی فازی نرمال و مجموعه‌ی فازی زیر نرمال: A را یک مجموعه‌ی فازی نرمال می‌نامند اگر ارتفاع آن یک باشد. اگر ارتفاع آن یک نباشد مجموعه‌ی فازی را زیر نرمال می‌نامند.

مجموعه‌های فازی در شکل ۲.۱ و در شکل ۳.۱ مجموعه‌های فازی نرمال هستند ولی مجموعه‌ی فازی شکل ۵.۱ یک مجموعه فازی زیر نرمال است.



شکل ۵.۱: یک مثال از مجموعه‌ی فازی زیر نرمال

تعریف ۷.۳.۱. هسته (مرکز) مجموعه‌های فازی: ما باید این مفهوم را برای چهار موقعیت مختلف تعریف کنیم. اگر تابع عضویت یک مجموعه‌ی فازی، حداکثر مقدار خود را تنها در یک عنصر از مجموعه‌ی جهانی بگیرد ،آن عضو را به‌عنوان هسته‌ی مجموعه‌ی فازی می‌نامند (شکل a ۶.۱). اگر تابع عضویت یک مجموعه‌ی فازی حداکثر مقدار خود را در بیش از یک عنصر از مجموعه‌ی جهانی می‌گیرد و تمام این عناصر کران‌دار باشند، نقطه میانی، عنصر مرکز است. (شکل b ۶.۱) اگر تابع عضویت یک مجموعه‌ی فازی حداکثر مقدار خود را در بیش از یک عنصر از مجموعه جهانی بگیرد و همه‌ی عناصر کران‌دار نباشند، بزرگ‌ترین عضو کران‌دار را مرکز می‌نامیم. (شکل d ۶.۱) در غیر این صورت، کوچک‌ترین عضو مرکز است.(شکل c ۶.۱)



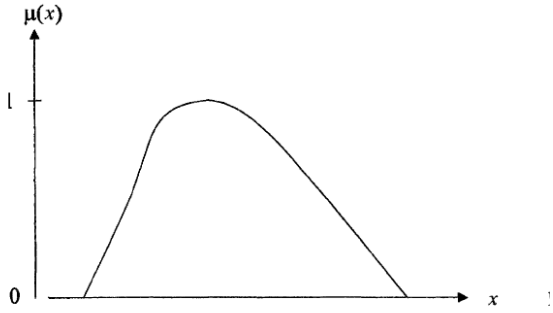
شکل ۶.۱: یک تعریف از مرکز مجموعه‌ی فازی برای چهار مورد مختلف

تعریف ۸.۳.۱. مجموعه‌های فازی محدب: مجموعه‌ی فازی $\tilde{A}$ که مجموعه‌ی جهانی آن $[a, b]$ هست، محدب است اگر و تنها اگر

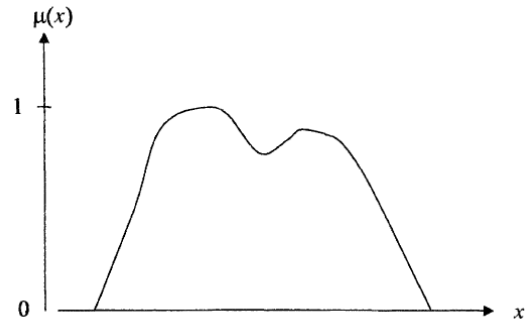
$$\mu_{\tilde{A}}(\lambda x_1 + (1-\lambda)x_2) \geq \min[\mu_{\tilde{A}}(x_1), \mu_{\tilde{A}}(x_2)]$$

$$\forall x_1, x_2 \in [a, b] \quad , \quad \forall \lambda \in [0, 1]$$

که min نشان می‌دهد عملگر حداقل ، از کوچک‌ترین مقدار عضویت از دو مقدار عضویت ، به‌عنوان نتیجه عمل استفاده می‌کند.
مجموعه‌ی فازی که در شکل ۷.۱ نشان داده‌شده است، محدب است درحالی‌که مجموعه‌ي فازي در شکل ۸.۱ محدب نیست.
در تعریف مجموعه فازی محدب، این نکته مهم است که نمودار توابع عضویت مجموعه فازی محدب، لزوماً محدب نیست. با وجود این، تعریف به توابع عضویت که مقعر باشد، نیاز دارد. البته طبق تعریف مجموعه فازی محدب، اگر تابع عضویت یک مجموعه فازی محدب باشد، مجموعه فازی محدب است. کنترل‌کننده‌ها و مدل‌های فازی از مجموعه فازی محدب استفاده می‌کنند. با توجه به تعریف مجموعه‌های فازی، هر تابع پیوسته یا گسسته می‌تواند یک تابع عضویت باشد تا زمانی که مقدار آن در $[0, 1]$ قرار بگیرد، بااین حال نوع گسسته رایج نیست.



شکل ۷.۱: یک مثال از تابع فازی محدب



شکل ۸.۱: یک مثال از تابع فازی غیر محدب

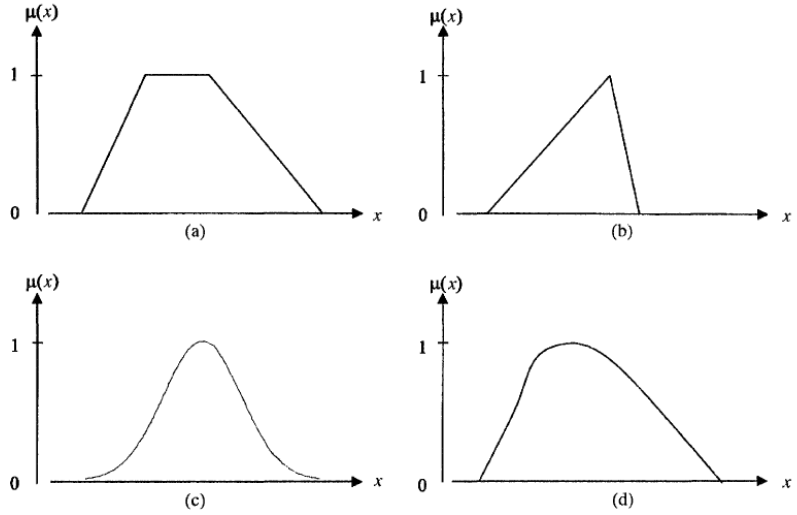
در واقع یکی از مسائل کلیدی در نظریه و کاربرد مجموعه‌های فازی، نحوه‌ی تعریف توابع عضویت مناسب در مجموعه‌های فازی است. رویکردهای اصلی عبارت‌اند از:

۱. درخواست متخصص کنترل و مدل‌سازی برای تعریف آن‌ها

۲. استفاده از داده‌های سیستم برای کنترل و مدل‌سازی آن‌ها

۳. ساختن آن به شیوه‌ی آزمون و خطا.

هر رویکرد متفاوت دارای مزایا و معایب است.
برنامه‌های کاربردی متعدد نشان داده‌اند که تنها چهار نوع از توابع عضویت در بیشتر شرایط موردنیاز است: ذوزنقه‌ای، مثلثی (یک مورد خاص از ذوزنقه‌ای)، گاوسی و زنگوله‌ای. شکل ۹.۱ برای هر کدام از موارد، یک مثال نشان داده‌شده است.
همه‌ی مجموعه‌های فازی ۹.۱ پیوسته، نرمال و محدب هستند. از میان چهار مورد، دو مورد اول کاربرد بیشتری دارند. در تصویر ما عمداً از توابع عضویت نامتقارن برای مثال‌های عمومی استفاده می‌کنیم، با این حال اغلب این‌طور نیست که همیشه از توابع متقارن استفاده شود.



شکل ۹.۱: مثال‌هایی از چهار مجموعه‌ی فازی a) ذوزنقه‌ای، b) مثلثی، c) منحنی گاوس، d) زنگوله‌ای شکل که همگی آن‌ها مجموعه‌های فازی پیوسته و نرمال و محدب هستند

۳.۳.۱ عملگرهای منطق فازی

در نظریه مجموعه‌های کلاسیک، عملگرهای منطقی دوتایی AND (اشتراک)، OR (اجتماع) و NOT (مکمل) وجود دارند. بصورت متناظر، عملگرهای منطق فازی در مجموعه‌های فازی نیز وجود دارد. اشتراك و اجتماع دو مجموعه فازي $\tilde{A}$ و $\tilde{B}$ به صورت زیر تعریف مي‌شود:

$$\mu_{\tilde{A}\cap\tilde{B}}(x) = \mu_{\tilde{A}}(x) \wedge \mu_{\tilde{B}}(x) = \min(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x))$$

$$\mu_{\tilde{A}\cup\tilde{B}}(x) = \mu_{\tilde{A}}(x) \vee \mu_{\tilde{B}}(x) = \max(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x))$$

عملگرهای منطق فازی AND و OR در کنترل‌کننده‌های فازی استفاده می‌شوند. برخلاف عملگرهای باینری AND و OR که به‌طور منحصربه‌فرد تعریف می‌شوند، عملگرهای فازی منحصربه‌فرد نیستند. عملگرهای فازی زاده [۱] و عملگرهای فازی لوکاسیویز [۲] به‌عنوان پرکاربردترین عملگرها برای کنترل فازی و مدل‌سازی مورد استفاده قرار مي‌گیرند و تعاریف آن‌ها به‌صورت زیر است:

۱. عملگر منطق فازی (AND) زاده

$$\mu_{\tilde{A}\cap\tilde{B}}(x) = \min(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x))$$

[1] Zadeh
[2] Lukasiewiez

۲. عملگر منطق فازی (AND) لوکاسیویز

$$\mu_{\tilde{A}\cap\tilde{B}}(x) = \mu_{\tilde{A}}(x) \cdot \mu_{\tilde{B}}(x)$$

۳. عملگر منطق فازی (OR) زاده

$$\mu_{\tilde{A}\cup\tilde{B}}(x) = \max(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x))$$

۴. عملگر منطق فازی (OR) لوکاسیویز

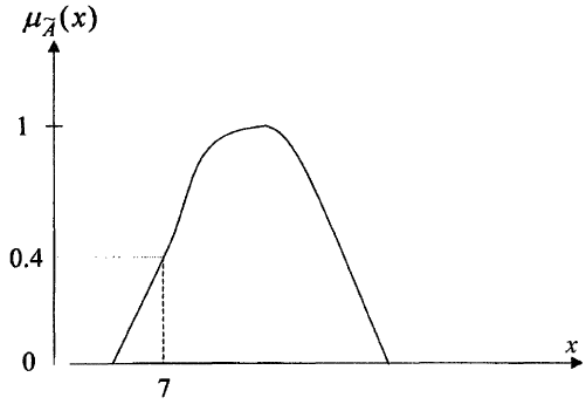
$$\mu_{\tilde{A}\cup\tilde{B}}(x) = \min(\mu_{\tilde{A}}(x) + \mu_{\tilde{B}}(x), ۱)$$

که $\max$ و $\min$ به ترتیب عملگر ماکزیمم و عملگر مینیمم است.

به‌عنوان یک اثبات پیوسته، فرض کنید که یک سن خاص، مثلاً سن ۳۰ سال با مقدار عضویت ۰.۸ «جوان» (یک مجموعه فازی) است و با مقدار عضویت ۰.۳ «پیر» (یک مجموعه فازی دیگر) است. بنابراین مقدار عضویت برای سن «جوان وپیر» (یک مجموعه فازی که اخیرا تشکیل‌شده است) مقدار ۰.۳ است ، اگر از عملگر فازی (AND) زاده استفاده شود و یا ۰.۲۴ است ، اگر از عملگر فازی (AND) لوکاسیویز استفاده شود. به همین ترتیب مقدار عضویت برای سن «جوان یا پیر» (مجموعه فازی جدید) ۰.۸ است ، اگر از عملگر فازی (OR) زاده استفاده شود و یا یک است اگر از عملگر فازی (OR) لوکاسیویز استفاده شود.

۴.۱ فازی سازی

کنترل فازی و مدل‌سازی همیشه شامل یک فرآیند در زمان است که فازی‌سازی نام دارد. فازی‌سازی یک روش ریاضی برای تبدیل یک عنصر از مجموعه جهانی و اختصاص دادن یک مقدار عضویت به آن در مجموعه فازی است. فرض کنید مجموعه‌ی فازی $\tilde{A}$ را روی $[a,b]$ تعریف می‌کنیم؛ یعنی $[a,b]$ مجموعه جهانی است و برای هر $x \in [a,b]$ فازی‌سازی به‌سادگی به‌صورت $\mu_{\tilde{A}}(x)$ است. شکل ۱۰.۱ یک مثال را نشان می‌دهد که نتیجه فازی‌سازی $\mu(۷) = ۰.۴$ است.



شکل ۱.۱۰: یک مثال برای نشان دادن فازی‌سازی

۵.۱ قوانین فازی

یک کنترل‌کننده فازی، از قوانین فازی استفاده می‌کند که وابسته به عبارات اگر–آن‌گاه است که شامل مجموعه‌های فازی، منطق فازی و استنتاج فازی است. قوانین فازی نقش کلیدی در ارائه دانش و تجربه کارشناسانه مدل و کنترل و ارتباط دادن متغیرهای ورودی کنترل‌کننده‌ی فازی و مدل‌سازی به متغیر خروجی دارند. دو نوع اصلی از قوانین فازی ، قوانین فازی ممدانی [۳] و قوانین فازی تاکاگی–سوگنو [۴] (TS) می‌باشند.

۱.۵.۱ قوانین فازی ممدانی

یک نمونه از قانون فازی ممدانی که حرکت یک ماشین را توضیح می‌دهد به‌صورت زیر است (اگر سرعت بالا باشد و شتاب کم باشد آن‌گاه ترمز باید نسبتاً کم باشد) که سرعت و شتاب متغیرهای ورودی هستند و ترمز کردن یک متغیر خروجی است. ’’ بالا ’’، ’’ کم’’ و ’’نسبتاً کم ’’ مجموعه‌های فازی هستند. دو مورد اول مجموعه‌ی فازی ورودی هستند درحالی که آخری مجموعه‌ی فازی خروجی نامیده می‌شود. متغیرها و همچنین عبارت‌های زبانی مانند ’’بالا ’’ می‌توانند با نمادهای ریاضی نشان داده شوند؛ بنابراین قوانین فازی ممدانی برای یک کنترل‌کننده‌ی فازی شامل سه متغیر ورودی و دو متغیر خروجی می‌تواند به شکل زیر توصیف شود:

(الف): «اگر $x_۱$، $\tilde{A}$ هست و $x_۲$، $\tilde{B}$ هست و $x_۳$، $\tilde{C}$ هست ، بنابراین $u_۱$، $\tilde{D}$ هست و $u_۲$، $\tilde{E}$ هست »

[3] Mamdani
[4] Takagi-Sugeno

که $x_۱$ و $x_۲$ و $x_۳$ متغیرهای ورودی هستند و $u_۱$ و $u_۲$ متغیرهای خروجی می‌باشند، این متغیرها می‌توانند به‌صورت پیوسته یا گسسته باشند. با این حال، عملاً تمام کنترل‌کننده‌های فازی و مدل‌ها با استفاده از کامپیوترهای دیجیتالی اجرا می‌شوند. $\tilde{A}$ و $\tilde{B}$ و $\tilde{C}$ و $\tilde{D}$ و $\tilde{E}$ مجموعه‌های فازی هستند و AND عملگر منطق فازی است. «$x_۱$، $\tilde{A}$ هست و $x_۲$، $\tilde{B}$ هست و $x_۳$، $\tilde{C}$ هست » به‌عنوان پیش شرط (قید) می‌نامیم و قسمت باقی‌مانده به‌عنوان نتیجه نامیده می‌شود. ساختار قوانین فازی ممدانی برای مدل‌سازی فازی یکسان است.
یک قانون فازی کلی برای کنترل فازی یا مدل‌سازی فازی می‌تواند به‌صورت زیر بیان شود:

(ب): اگر $v_۱$ در $\tilde{S}_۱$ و ... و v_M در $\tilde{S}_M$ باشد آن‌گاه $Z_۱$ در $\tilde{W}_۱$ و ... Z_p در $\tilde{W}_p$ است.

که در آن v_i، $i = ۱,\dots M$ متغیر ورودی و Z_j، $j = ۱,\dots,P$ متغیر خروجی است. $\tilde{S}_i$ مجموعه‌ی فازی ورودی و $\tilde{W}_j$ مجموعه‌ی فازی خروجی می‌باشد.
همان‌طور که قبلا اشاره شد، برای بیشتر کنترل‌کننده‌های فازی و مدل‌ها، مجموعه‌های فازی ورودی پیوسته، نرمال و محدب هستند و معمولا از چهار نوع رایج می‌باشند. مجموعه‌های فازی خروجی اغلب از نوع یکتا هستند؛ بنابراین قانون کلی فازی ممدانی ۱.۵.۱ می‌تواند به‌صورت زیر کاهش پیدا کند:

(پ): اگر $v_۱$ در $\tilde{S}_۱$ و ... و v_M در $\tilde{S}_M$ بنابراین $Z_۱$ در $\beta_۱$ و ... و Z_p در β_p هست.

که در آن β_j اعضای مجموعه‌های فازی یکتا $\tilde{W}_j$ که غیر صفر می‌باشند فقط درصورتی‌که $Z_j = \beta_j$.

۲.۵.۱ قوانین فازی TS

حال به قوانین فازی TS می‌پردازیم. برخلاف قوانین فازی ممدانی، قانون TS از توابع متغیرهای ورودی به‌عنوان قانون نتیجه استفاده می‌کند. برای کنترل فازی، قانون TS متناظر با قانون ممدانی ۱.۵.۱ به صوت زیر است

(الف): اگر $x_۱$ در $\tilde{A}$ و $x_۲$ در $\tilde{B}$ و $x_۳$ در $\tilde{C}$ باشد ، آن‌گاه

$$u_۱ = f(x_۱, x_۲, x_۳) \qquad \text{و} \qquad u_۲ = g(x_۱, x_۲, x_۳)$$

که f و g توابع حقیقی از هر نوع هستند. به‌طور مشابه برای مدل‌سازی فازی، قانون TS متناظر با قانون ممدانی ۱.۵.۱ به‌صورت زیر است.

(ب): اگر $y(n)$ در $\tilde{A}$ و $y(n-۱)$ در $\tilde{B}$ و $y(n-۲)$ در $\tilde{C}$ و $u(n)$ در $\tilde{D}$ و $u(n-۱)$ در $\tilde{E}$ باشد، آن‌گاه

$$y(n+۱) = F(y(n), y(n-۱), y(n-۲), u(n), u(n-۱))$$

که F یک تابع دلخواه است. مطابق با قانون کلی ممدانی ۱.۵.۱ یک قانون کلی TS برای کنترل فازی و مدل‌سازی به‌صورت زیر است:

(پ): اگر $v_۱$ در $\tilde{S}_۱$ و ... و v_M در $\tilde{S}_M$ آن‌گاه

$$Z_۱ = f_۱(v_۱, \ldots, v_M), \ldots, Z_p = f_p(v_۱, \ldots, v_M)$$

در این نظریه f_j می‌تواند تابع حقیقی، خطی یا غیرخطی باشد . به نظر می‌رسد استفاده از توابع غیرخطی برای تمام قوانین و یا استفاده از ترکیب خطی و غیرخطی به‌عنوان نتیجه قوانین (به‌عنوان‌مثال، توابع خطی برای برخی قوانین و توابع غیرخطی برای باقی‌مانده) در این روش ، کلي‌تر و قوي‌تر باشند. متاسفانه این ایده غیرعملی است زیرا انتخاب توابع غیرخطی برای هر قانون فازی، غیرممکن و یا بسیار دشوار است. این مشکل اساسا به همان اندازه‌ای است که در نظریه مدل‌سازی کنترل غیرخطی کلاسیک با آن روبرو می‌شویم. به همین دلیل، توابع خطی در پژوهش‌های نظری و توسعه‌ی عملی کنترل‌کننده‌ها و مدل‌های فازی TS مورد استفاده قرار گرفته‌اند. در این پایان‌نامه فقط بر روی کنترل‌کننده‌ها و مدل‌های فازی که از قانون TS خطی استفاده می‌کنند تمرکز می‌کنیم.

۶.۱ استنتاج فازی

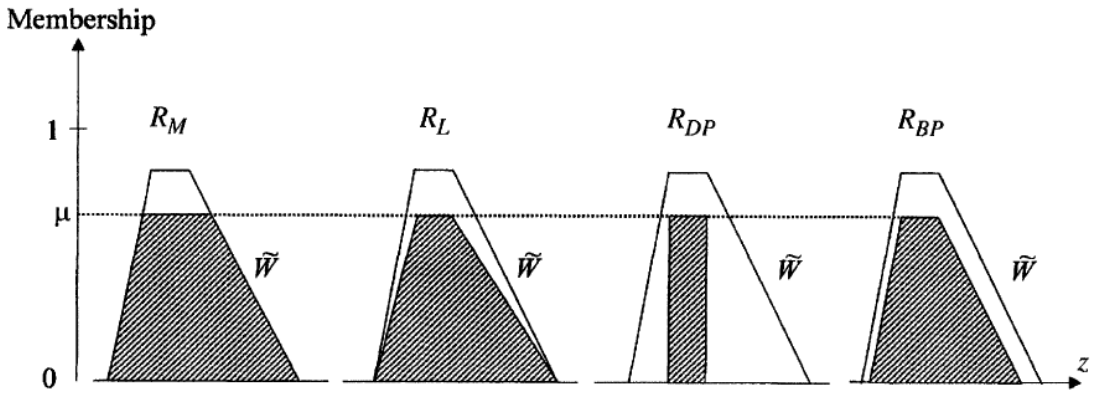
استنتاج فازی گاهی اوقات استدلال فازی یا استدلال تقریبی نامیده می‌شود و برای به دست آوردن خروجی از قوانین داده شده توسط اطلاعات ورودی استفاده می‌شود. قوانین فازی، استراتژی کنترل یا دانش یا تجربه مدل‌سازی را نشان می‌دهند. هنگامی‌که اطلاعات خاص به متغیرهای ورودی در پیش‌شرط اختصاص داده می‌شود، استنتاج فازی برای محاسبه نتیجه برای متغیرهای خروجی در قانون نتیجه موردنیاز است. قوانین فازی ممدانی و قوانین فازی TS از روش‌های استنتاج فازی مختلفی استفاده می‌کنند. برای قوانین کلی فازی ممدانی، سؤال در مورد استنتاج فازی به‌صورت زیر است؛ با توجه به $v_i = \alpha_i$ برای هر i که اعداد α_i حقیقی هستند، Z_j چه باید باشد؟

برای کنترل فازی و مدل‌سازی، پس از فازی سازی v_i به α_i و به‌کاربردن عملگرهای منطق فازی AND روی مقدار عضویت ، نتیجه در قانون فازی یک ترکیب از مقدار عضویت μ که خروجی برای قوانین پیش‌شرط است را به دست می‌آوریم. سپس این سؤال پیش می‌آید که

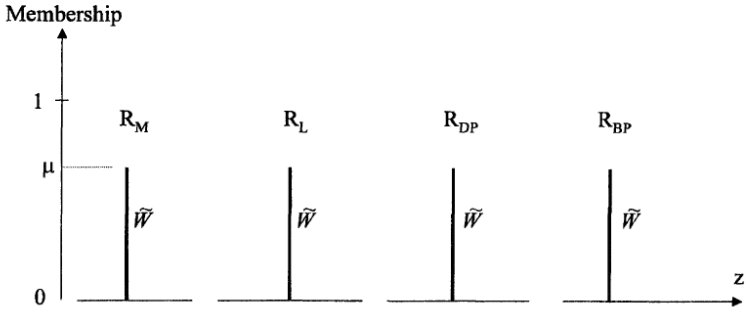
چگونه '' آن‌گاه '' را در قانون نتیجه محاسبه کنیم؟ محاسبه کردن آن‌گاه '' استنتاج فازی '' نامیده می‌شود. به‌طور واضح، سؤال این است که فرض کنید که μ داده‌شده باشد چگونه باید Z_j محاسبه شود؟ از آنجایی‌که از حیث ریاضیاتی محاسبه متغیرهای خروجی مختلف یکسان است، ما به ترتیب از Z و $\tilde{W}$ استفاده می‌کنیم تا z_j و $\tilde{W}_j$ را در بحث زیر روی روش‌های استنتاج فازی نشان دهیم. تعدادی از روش‌های استنتاج فازی می‌تواند برای به انجام رساندن این امر استفاده شود؛ اما فقط چهار مورد از آن‌ها در مدل‌سازی و کنترل فازی عمومی هستند و ما فقط در این پایان‌نامه از آن‌ها استفاده خواهیم کرد. از جمله این روش‌ها ، روش استنتاج مینیمم ممدانی، روش استنتاج حاصل‌ضرب لارسن، روش استنتاج حاصل‌ضرب دراستیک و روش استنتاج حاصل‌ضرب کران‌دار هستند، ما آن‌ها را به ترتیب با R_M، R_L، R_{DP} و R_{BP} نشان می‌دهیم. تعاریفی از این روش‌ها در جدول ۱.۱ ارائه شده است که $\mu_{\tilde{W}}(z)$ یک تابع عضویت از مجموعه‌ی فازی $\tilde{W}$ در قانون فازی ۱.۵.۱ و μ عضو ترکیب‌شده در پیش‌شرط است. برای فهم بهتر ما به‌طور گرافیکی تعاریفی در شکل ۱۱.۱ نشان می‌دهیم. نتایج چهار روش استنتاج فازی ، مجموعه‌های فازی تشکیل‌شده به‌وسیله‌ی ناحیه‌های سایه‌دار هستند. به‌وضوح نتیجه مجموعه‌های فازی می‌تواند سریع به دست آید. فرمول‌های توصیف‌کننده نواحی سایه‌دار می‌تواند به‌طور ریاضیاتی محاسبه شوند. در بین چهار روش، روش ممدانی به‌طور گسترده‌ای در مدل‌سازی و کنترل فازی استفاده می‌شود.

جدول ۱.۱: تعاریفی از چهار روش استنتاج برای کنترل فازی و مدل‌سازی (الف).استنتاج مینیمم ممدانی، (ب).استنتاج حاصل‌ضرب لارسن، (ج).استنتاج حاصل‌ضرب دراستیک، (د).استنتاج حاصل‌ضرب کران‌دار

روش استنتاج فازی	تعریف
R_M: استنتاج مینیمم ممدانی	برای هر z، $\min(\mu, \mu_{\tilde{w}}(z))$
R_L: استنتاج حاصل‌ضرب لارسن	برای هر z، $\mu \cdot \mu_{\tilde{w}}(z)$
R_{DP}: استنتاج حاصل‌ضرب دراستیک	$\begin{cases} \mu, & \mu_{\tilde{W}}(z) = ۱ \\ \mu_{\tilde{w}}(z), & \mu = ۱ \\ ۰, & \mu < ۱, \mu_{\tilde{w}}(z) < ۱ \end{cases}$
R_{BP}: استنتاج حاصل‌ضرب کران‌دار	$\max(\mu + \mu_{\tilde{w}}(z) - ۱, ۰)$



شکل ۱۱.۱: توضیح تصویری از تعاریف چهار روش استنتاج فازی که در جدول ۱.۱ ارائه‌شده است. (الف).استنتاج مینیمم ممدانی، (ب).استنتاج حاصل‌ضرب لارسن، (ج).استنتاج حاصل‌ضرب دراستیک، (د).استنتاج حاصل‌ضرب کران‌دار



شکل ۱۲.۱: برای مدل‌ها و کنترل‌کننده‌های فازی ممدانی از مجموعه‌های فازی یکتا به‌عنوان قانون نتیجه استفاده می‌کنیم. نتیجه‌ی استفاده از چهار روش استنتاج مختلف برابر است

کنترل‌کننده‌های فازی ممدانی ، مجموعه‌های فازی یکتا را به‌عنوان نتیجه به‌کار می‌برند (۱.۵.۱). تحت این شرایط، چهار روش مختلف استنتاج، همان‌طور که در شکل ۱۲.۱ نشان داده‌شده است ، نتایج استنتاج یکسانی ایجاد می‌کنند.

برای قانون فازی TS، استنتاج فازی ساده‌تر است و تنها یک روش وجود دارد. برای قانون کلی فازی TS ۲.۵.۱ نتیجه استنتاج فازی برای Z_j ، به‌صورت $\mu \cdot f_j(v_1, \ldots, v_M)$ است .

۷.۱ غیرفازی‌سازی

غیرفازی‌سازی یک پردازش ریاضی است که از آن برای تبدیل مجموعه فازی به یک عدد حقیقی استفاده می‌شود. مجموعه‌های فازی که به‌وسیله استنتاج فازی در قوانین فازی تولید می‌شوند باید به نحوی ترکیب شوند تا یک عدد، به‌عنوان خروجی یک مدل یا کنترل‌کننده فازی به‌دست آید . محرک‌ها برای سیستم‌های کنترل می‌توانند فقط یک مقدار را به‌عنوان سیگنال ورودی‌شان بپذیرند. هر مدل و کنترل‌کننده فازی از یک غیرفازی کننده استفاده می‌کند که یک فرمول ریاضی ساده است. برای مدل‌ها و کنترل‌کننده‌های فازی با بیشتر از یک متغییر خروجی، غیرفازی‌سازی برای هر یک از آن‌ها به‌طور مجزا اما با یک روش مشابه اعمال می‌شود. در بیشتر موارد فقط یک غیرفازی‌کننده برای همه‌ی متغییرهای خروجی استفاده می‌شود.
انواع متفاوتی از غیرفازی‌کننده‌ها وجود دارند که برای رویدادهای مختلف مناسب هستند. در ادامه تعدادی از غیرفازی‌کننده‌های محبوب‌تر را بررسی می‌کنیم. از آن‌جا که بیش‌تر مدل‌ها و کنترل‌کننده‌های فازی از مجموعه‌های فازی یکتا در قانون فازی نتیجه استفاده می‌کنند، ما روی مجموعه‌های فازی خروجی یکتا تمرکز خواهیم کرد.

۱.۷.۱ تعمیم غیرفازی‌سازی

غیرفازی‌سازی تعمیم‌یافته، تعداد مختلفی از غیرفازی‌کننده‌ها را در یک فرمول ساده ریاضی نمایش می‌دهد. فرض کنید z که متغییر خروجی یک مدل یا کنترل‌کننده‌ی فازی است و ارزیابی N قانون فازی ممدانی با استفاده از بعضی روش‌های استنتاج فازی، N مقدار عضویت، $\mu_1,\ldots,\mu_N$ را برای N مجموعه‌ی فازی خروجی یکتا در قوانین تولید می‌کند (یک مقدار برای هر قانون). فرض کنید این مجموعه‌های فازی فقط در $z = \beta_1,\ldots,\beta_N$ غیر صفر (مثبت) هستند. غیرفازی‌سازي تعمیم‌یافته نتایج غیرفازی‌سازی زیر را تولید می‌کنند:

$$z = \frac{\sum_{k=1}^{N} \mu_k^{\alpha}.\beta_k}{\sum_{k=1}^{N} \mu_k^{\alpha}}, \qquad (۱.۱)$$

که α یک پارامتر طراحی است. در ادامه موارد بالا، فرض کنید که مدل‌ها و کنترل‌کننده‌های فازی از قانون TS استفاده می‌کنند.
فرض کنید که قانون نتیجه، N قانون فازی $g_k(v_1,\ldots,v_M)$، $k = 1,\ldots,N$ باشند؛ بنابراین خروجی غیرفازی‌سازی با استفاده از غیرفازی‌سازی تعمیم‌یافته به شکل زیر به دست می‌آید:

$$z = \frac{\sum_{k=1}^{N} \mu_k^{\alpha} \cdot g_k(v_1,\ldots,v_M)}{\sum_{k=1}^{N} \mu_k^{\alpha}} \qquad (۲.۱)$$

۲.۷.۱ غیرفازي‌کننده خطي با مرکز ثقل

انواع مختلفی از غیرفازی‌کننده‌ها هستند که از مقادیر مختلف α در تعمیم غیرفازی کردن استفاده می‌کنند که $۰ \leq \alpha < \infty$. زمانی‌که $\alpha = ۱$، از مرکز ثقل به‌عنوان غیرفازی‌کننده استفاده می‌شود زیرا مرکز ثقل، مجموعه‌های فازی یکتا از قوانین مختلف را محاسبه می‌کند. گاهی از معنی حداکثر غیرفازی‌سازی در مواقعی که $\alpha = \infty$ درنظرگرفته می‌شود استفاده می‌کنیم. زمانی‌که از قانون ممدانی استفاده می‌کنیم، نتیجه غیرفازی‌سازی به‌صورت زیر است:

$$z = \sum_{k=۱}^{N} \mu_k^{\alpha} \cdot \beta_k. \qquad (۳.۱)$$

از طرفي ، برای قوانین فازی TS، داریم:

$$z = \sum_{k=۱}^{N} \mu_k^{\alpha} \cdot g_k(v_۱, \ldots, v_M)$$

۸.۱ مثال‌هایی از کنترل‌کننده‌ی فازی

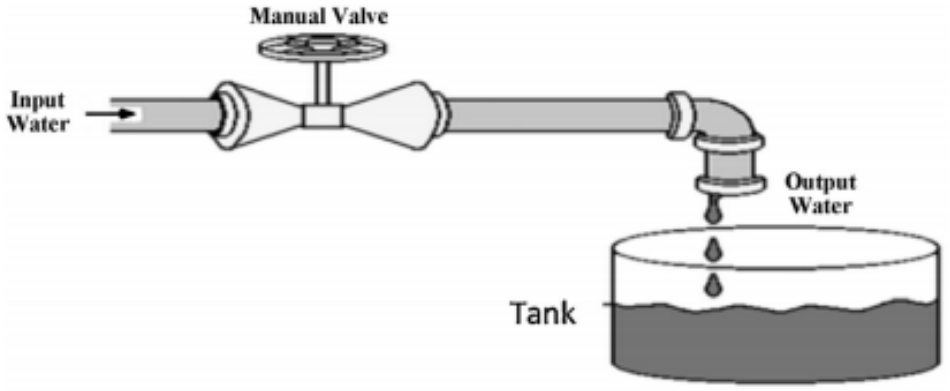
در این بخش نمونه‌هایی از کاربرد کنترل‌کننده فازی را مورد بررسی قرار می‌دهیم.

۱.۸.۱ کنترل‌کننده فازی برای مخزن آب

هدف کنترل‌کننده ، یک سطح خاصی در مخزن است. بنابراین ما باید سطح واقعي آب در مخزن را بدانیم و با وجود آن باید بتوانیم شیرها را تنظیم کنیم. شکل۱۳.۱ به‌صورت گرافیکی روش عملکرد باز شدن شیر و روند پر شدن مخزن را نشان می‌دهد. دو متغیر داریم که سطح آب به‌عنوان ورودی و سرعت باز شدن شیر مخزن، خروجی است.

هدف اصلی در این حالت طراحی کنترل‌کننده فازی برای حفظ سطح آب در مخزن است. همان‌طور که در شکل ۱۴.۱ نشان داده‌شده است کنترل‌کننده داراي دو ورودی برای سیستم فازی است . اولین ورودی سطح آب است که سه تابع عضویت با مقادیر کیفي ’’زیاد’’، ’’خوب ’’ و ’’کم’’ دارد. دومین متغیر ورودی ، نرخ نامیده می‌شود که داراي سه تابع عضویت با مقادیر کیفی ’’منفی ’’، ’’خوب ’’ و ’’مثبت ’’ است. نام برچسب‌های کیفی ، بر اساس فرآیندهای تجربی رفتارهای پر شدن مخزن آب تنظیم شده‌اند. سیستم استنتاج فازی دارای یک متغیر خروجی به نام ’’شیر ’’ است که دارای پنج تابع عضویت ’’ مثلثی ’’ با مقادیر کیفی ’’بستن سریع ’’، ’’بستن آرام ’’، ’’بدون تغییر ’’، ’’باز شدن آرام ’’و ’’باز شدن سریع ’’است. قوانین کنترل‌کننده فازی به‌صورت زیر هستند:

۱. اگر (سطح خوب باشد) آنگاه (شیر بدون تغییر می‌ماند)



شکل ۱۳.۱: نمایش گرافیکی مسئله کنترل‌کننده فازی مخزن آب

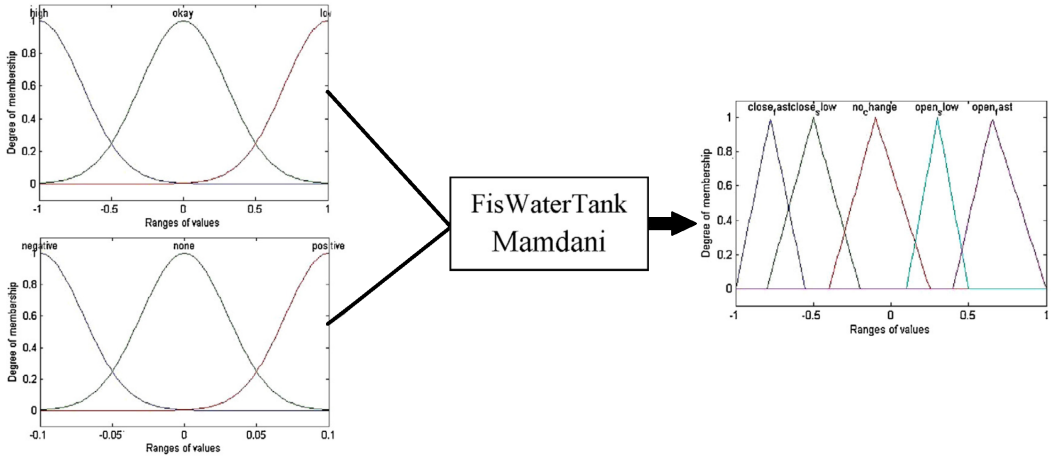
۲. اگر (سطح کم باشد) آنگاه (شیر سریع باز می‌شود)

۳. اگر (سطح زیاد باشد) آنگاه (شیر سریع بسته می‌شود)

۴. اگر (سطح خوب باشد) و (نرخ مثبت باشد) و آنگاه (شیر آرام بسته می‌شود)

۵. اگر (سطح خوب باشد) و (نرخ منفی باشد) و آنگاه (شیر آرام باز می‌شود)

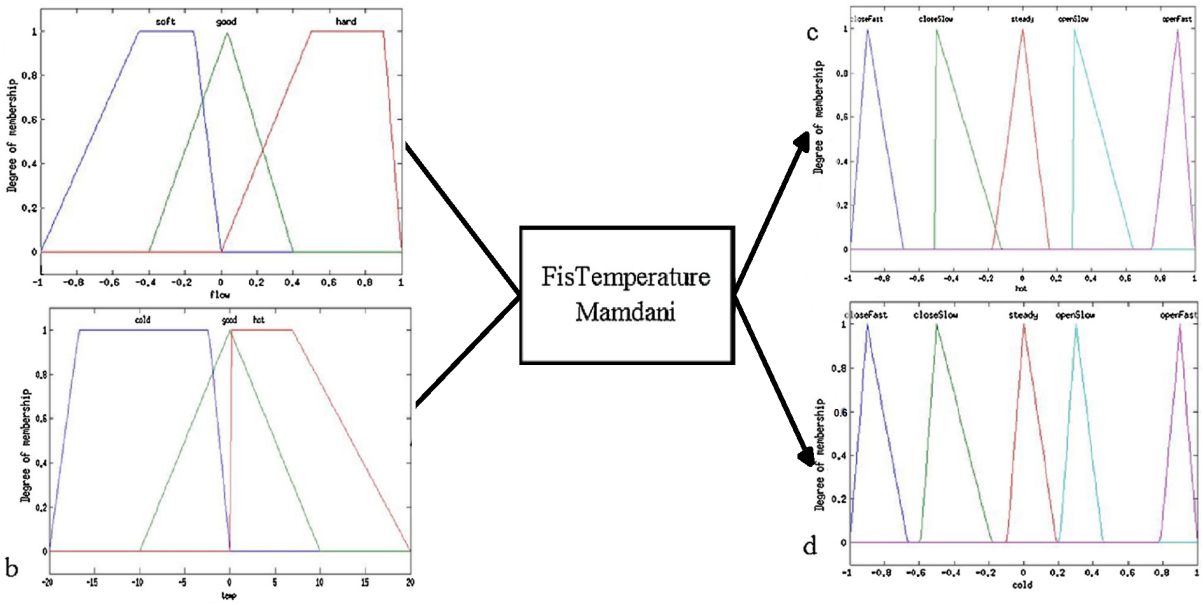
ترتیب نقش‌ها بر اساس این‌که چگونه فرآیند پرکردن مخزن آب عمل می‌کند، به‌طور تجربی پیدا شده است. ما این ۵ قانون را برای نمایش رفتار کنترل‌کننده فازی در نظر می‌گیریم.



شکل ۱۴.۱: ساختار عمومی سیستم فازی با متغیرهای ورودی و خروجی

۲.۸.۱ مسئله کنترل دما

در این بخش مسئله کنترل دما در جریان آب را مورد مطالعه قرار می‌دهیم. کنترل‌کننده این مسئله از نوع ممدانی است و پارامترهای ورودی و خروجی توسط متغیرهای کیفی نمایش داده شده‌اند. مقادیر کیفی متغیرهاي ورودی ’’دما’’و ’’جریان’’ و متغیرهاي خروجی ’’سرد’’ و ’’گرم’’ در شکل ۱۵.۱ نشان داده شده‌اند.



شکل ۱۵.۱: (a)جریان،(b)دما،(c)گرم،(d)سرد

تابع عضویت برای متغیرهاي ورودی از نوع ذوزنقه‌ای و مثلثی هستند و تابع عضویت متغیرهای خروجي از نوع مثلثی هستند. ۹ قانون استنتاج تعریف شده برای این کنترل‌کننده عبارت‌اند از:

۱. اگر (دما سرد باشد) و (جریان نرم باشد) آنگاه (سرد آرام باز مي‌شود) و (گرم سریع باز مي‌شود).

۲. اگر (دما سرد باشد) و (جریان خوب باشد) آنگاه (سرد آرام بسته مي‌شود) و (گرم آرام باز مي‌شود).

۳. اگر (دما سرد باشد) و (جریان سخت باشد) آنگاه (سرد سریع بسته مي‌شود) و (گرم آرام بسته مي‌شود).

۴. اگر (دما خوب باشد) و (جریان نرم باشد) آنگاه (سرد آرام باز مي‌شود) و (گرم آرام باز مي‌شود).

۵. اگر (دما خوب باشد) و (جریان خوب باشد) آنگاه (سرد یکنواخت است) و (گرم یکنواخت است).

۶. اگر (دما خوب باشد) و (جریان سخت باشد) آنگاه (سرد آرام بسته مي‌شود) و (گرم آرام باز مي‌شود).

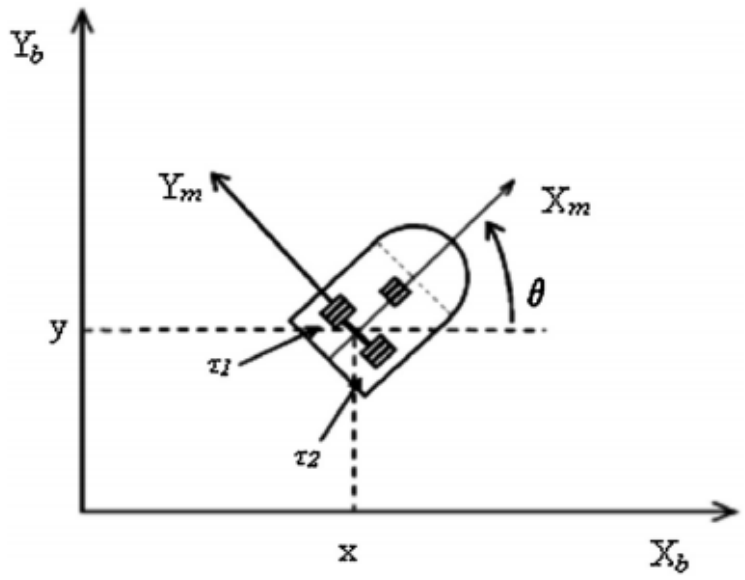
۷. اگر (دما گرم باشد) و (جریان نرم باشد) آنگاه (سرد سریع باز مي‌شود) و (گرم آرام باز مي‌شود).

۸. اگر (دما گرم باشد) و (جریان خوب باشد) آنگاه (سرد آرام باز مي‌شود) و (گرم آرام بسته مي‌شود).

۹. اگر(دما گرم باشد) و (جریان سخت باشد) آنگاه (سرد آرام بسته مي‌شود) و (گرم سریع بسته مي‌شود)

۳.۸.۱ ربات متحرک خودکار

منظور از ربات متحرک، یک دستگاه خودکار است که قادر به پیگیری مسیرهای قابل پیش‌بینی در محیط باشد و در شکل۱۶.۱ نشان داده شده است.



شكل ۱۶.۱: يك نمونه ربات متحرك به همراه پارامترهاي ورودي

بدنه ربات در اطراف محور عمودی متقارن است و مرکز جرم در مرکز هندسی بدنه قرار دارد. دو چرخ محرک دارد که در محوری که از مرکز جرم "C" می‌گذرد ثابت شده‌اند و توسط

$\{C, X_m, Y_m\}$ نشان داده می‌شوند و یک چرخ غیرفعال با حرکت طبق برنامه ربات را از سرگردان شدن باز می‌دارد . در [۳۱، ۳۲] دینامیک ربات متحرک توسط معادله (۴.۱) و (۵.۱) ارائه شده است.

$$M(q)\dot{v} + C(q,\dot{q})v + Dv = \tau + P(t) \qquad (۴.۱)$$

که $q = (x, y, \theta)^T$ بردار مختصات پیکربندی است.

$v = (v, w)^T$ بردار سرعت‌ها هستند.

$\tau = (\tau_۱, \tau_۲)$ بردار گشتاور اعمال‌شده به چرخ‌های ربات است که $\tau_۱$ و $\tau_۲$ به ترتیب گشتاور راست و چپ چرخ‌ها هستند.

$M(q) \in R^{۲\times۲}$ یک ماتریس اینرسی مثبت است. $P \in R^۲$ یک بردار مختلط محدود است.

$C(q,\dot{q})$ بردار نیروهای سانتریفیوژ و کلیتیوس است و $D \in R^{۲\times۲}$ یك ماتریس استهلاك مثبت است.

سیستم سینماتیکی توسط معادله زیر نمایش داده می‌شود.

$$\dot{q} = \underbrace{\begin{bmatrix} \cos\theta & ۰ \\ \sin\theta & ۰ \\ ۰ & ۱ \end{bmatrix}}_{J(q)} \underbrace{\begin{bmatrix} v \\ w \end{bmatrix}}_{v} \qquad (۵.۱)$$

که (x, y) موقعیت ربات، θ زاویه بین جهت حرکت و محور x، v و w سرعت‌های خطی و زاویه‌ای هستند.

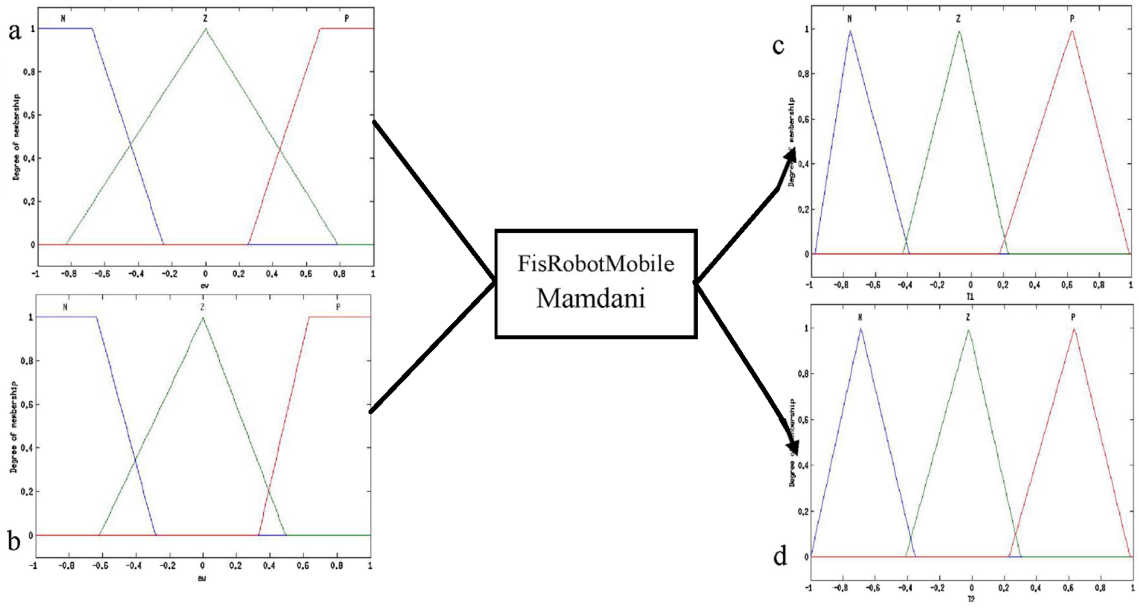
علاوه بر این، معادله (۶.۱) محدودیت‌های این سیستم را نشان می‌دهد که به شرایط یک چرخ غیر لغزشی مرتبط است که ربات را از حرکت به سمت جانب باز می‌دارد.

$$\dot{y}\cos\theta - \dot{x}\sin\theta = ۰ \qquad (۶.۱)$$

کنترل‌کننده از نوع ممدانی است و پارامترهای ورودی و خروجی توسط متغیرهای کیفی ارائه شده‌اند. متغیرهای ورودی ، خطا در سرعت خطی (e_v) و خطا در سرعت زاویه‌ای (e_w) است و متغیرهاي خروجی ، گشتاور راست $(\tau_۱)$ و گشتاور چپ $(\tau_۲)$ است که در شکل ۱۷.۱ نشان داده شده‌اند.

تابع عضویت متغیرهای ورودی و خروجی از نوع ذوزنقه‌ای در ترم‌های کیفی منفی (N) و مثبت (P) و مثلثی برای ترم کیفی صفر (Z) هستند. برای متغیر خروجی ما سه تابع عضویت، منفی (N) ، مثبت (P) ، صفر (Z) از نوع مثلثی داریم. محدوده $[۰, ۱]$ برای هر متغیر استفاده شده است زیرا تمام متغیرها را نرمال‌سازی کرده‌ایم. ۹ قانون فازی که استفاده‌شده‌اند عبارت‌اند از:

۱. اگر (خطاي سرعت خطي منفي باشد) و (خطاي سرعت زاويه اي منفي باشد) آنگاه (گشتاور راست منفي) و (گشتاور چپ منفي است).



شکل ۱۷.۱: (a)خطای سرعت خطی (e_v)،(b)خطای سرعت زاویه‌ای(e_w)،(c) گشتاور چپ(τ_1)،(d)گشتاور راست τ_2

۲. اگر (خطاي سرعت خطي منفي باشد) و (خطاي سرعت زاويه اي صفر باشد) آنگاه (گشتاور راست منفي) و (گشتاور چپ صفر است).

۳. اگر (خطاي سرعت خطي منفي باشد) و (خطاي سرعت زاويه‌اي مثبت باشد) آنگاه (گشتاور راست منفي) و (گشتاور چپ مثبت است).

۴. اگر (خطاي سرعت خطي صفر باشد) و (خطاي سرعت زاويه‌اي منفي باشد) آنگاه (گشتاور راست صفر) و (گشتاور چپ منفي است).

۵. اگر (خطاي سرعت خطي صفر باشد) و (خطاي سرعت زاويه‌اي صفر باشد) آنگاه (گشتاور راست صفر) و (گشتاور چپ صفر است).

۶. اگر (خطاي سرعت خطي صفر باشد) و (خطاي سرعت زاويه‌اي مثبت باشد) آنگاه (گشتاور راست صفر) و (گشتاور چپ مثبت است).

۷. اگر (خطاي سرعت خطي مثبت باشد) و (خطاي سرعت زاويه‌اي منفي باشد) آنگاه (گشتاور راست مثبت) و (گشتاور چپ منفي است).

۸. اگر (خطاي سرعت خطي مثبت باشد) و (خطاي سرعت زاويه‌اي صفر باشد) آنگاه (گشتاور راست مثبت) و (گشتاور چپ صفر است).

۹. اگر (خطاي سرعت خطي مثبت باشد) و (خطاي سرعت زاويه‌اي مثبت باشد) آنگاه (گشتاور راست مثبت) و (گشتاور چپ مثبت است).

فصل ۲

مسئله مسیریابي حرکت ربات

۱.۲ مقدمه

مسئله موردبحث در این پایان‌نامه، یافتن مسیر بهینه بین دو ایستگاه برای یک ربات یا گروهی از ربات‌هاست. این مسیر با توجه به هردو پارامتر مسافت و/یا زمان بهینه می‌شود و باید اطمینان حاصل شود که با یکدیگر یا با مانع که خود می‌تواند ثابت یا متحرک باشد، برخورد نکنند. رویکرد کلاسیک ممانعت از برخورد، بخصوص درزمینه‌ی برنامه‌ریزی حرکت ربات، استفاده از تابع پتانسیل است. سایر تکنیک‌های توسعه‌یافته شامل رویکردهایی از جمله الگوریتم‌های تصادفی و تکنیک‌های برنامه‌ریزی مسیر با استفاده از جستجوی گراف و پوشش سطح است. در این‌جا از برنامه‌ریزی عدد صحیح به‌عنوان یک چارچوب مدل‌سازی برای فرمول بندی مانع و ضوابط ممانعت از برخورد بین وسیله متحرک در سیستم پویا استفاده می‌کنیم.

دراین فصل فرمول‌بندی مسائل اساسی برای یک ربات ارائه می‌شود که توسط معرفی محدودیت‌ها برای اجتناب از مانع دنبال می‌شود. توجه داشته باشید که ما بین ممانعت از برخورد با موانع ثابت [۲]، ممانعت از برخورد با موانع متحرک که مسیرهای از پیش تعیین‌شده دارند [۳] و ممانعت از برخورد متقابل بین ربات‌های چندگانه [۴]، تفاوت قائل هستیم.

۲.۲ فرمول‌بندی مسئله

۱.۲.۲ فرمول‌بندی برای تک ربات

مسئله پایه‌ای برنامه‌ریزی مسیر که در این فصل بحث می‌شود، یافتن یک مسیر بهینه بین دو نقطه است که یک یا تعدادی ربات در حال حرکت هستند. راهکار کلاسیک برای این مسئله‌ی بهینه‌سازی شامل کمینه کردن تابع هزینه درجه دوم است که متغیرها باید فضای معادلات سیستم پویای (A_c, B_c) را ارضا کند [۵] و به صورت زیر بیان می‌شود:

$$\min_{s,u} J = \min_{s,u} \int_{\circ}^{\infty} (s'Qs + u'Ru)\mathrm{d}t \qquad (۱.۲)$$

مشروط براینکه

$$\dot{s} = A_c s + B_c u \qquad (۲.۲)$$

که در آن $s \in \mathbb{R}^n$ بردار حالت و $u \in \mathbb{R}^{n_u}$ بردار کنترل (بردار ورودی) است. فرض بر این است که سیستم از چندین حالت اصلی $s_\circ$ شروع به کار کند. حالت مقصد به‌طور ضمنی صفر تعریف شده است.

در این‌جا از نرم‌ـیك براي تابع هزینه استفاده می‌کنیم. ماتریس‌های وزنی مثبت Q و R فرمول بندی درجه دو توسط بردارهای وزنی غیر صفر و مثبت q و r جایگزین شده‌اند که تابع هزینه محدب زیر را نتیجه می‌دهد:

$$J = \int_{\circ}^{\infty} (q'|s| + r'|u|)\mathrm{d}t \qquad (۳.۲)$$

که $|u|$ بردار قدر مطلق u است. هنگامی‌که با محدودیت‌های ممانعت از برخورد ترکیب می‌شود (بخش (۳.۲) را ببینید) فرمول‌بندی نرم یک، یک مسئله خطی صحیح آمیخته می‌دهد که برای حل بسیار آسان‌تر از مسئله درجه دومی صحیح آمیخته است که با استفاده از تابع هزینه درجه دوم به‌دست‌آمده است.

به‌منظور حل عددی این مسئله کنترل بهینه، باید سیستم را تقریب بزنیم و از محدوده زمانی محدود T استفاده کنیم. محدوده T به گام‌های زمانی مجزای $N = T/\Delta t$ تبدیل می‌شود که Δt زمان نمونه است. برای زمان باقی‌مانده $T \to \infty$، یک هزینه نهایی $f(s_N)$ معرفی‌شده است. درنتیجه مسئله بهینه‌سازی به صورت زیر تبدیل می‌شود:

$$\min_{s_i,u_i} J_T = \min_{s_i,u_i} \sum_{i=\circ}^{N-۱} (q'|s_i| + r'|u_i|\Delta t + f(s_N)) \qquad (۴.۲)$$

باتوجه به

$$s_{i+۱} = As_i + Bu_i \qquad (۵.۲)$$

که A و B از A_c و B_c از بین تقریب‌های مناسب، مشتق شده است و بعلاوه با نادیده گرفتن عبارت ثابت $q'|s_0|$ و تقسیم تابع هزینه بر Δt، داریم:

$$J_T = \sum_{i=1}^{N-1} q'|s_i| + \sum_{i=0}^{N-1} r'|u_i| + f(s_N) \tag{۶.۲}$$

یک انتخاب ساده برای هزینه نهایی $f(s_N)$، $\mathbf{P}'|s_N|$ است که بردار وزن غیر منفی P می‌تواند به‌طور مناسبی انتخاب شود. این منحنی تابع هزینه خطی‌ای را می‌دهد که می‌تواند با معرفی متغیرها و محدودیت‌های اضافی [۶] به یک شکل خطی تبدیل شود.
با معرفی این متغیرهای اضافی $w_{ij}(i = 1,\dots,N, j = 1,\dots,n)$ و $v_{ik}(i = 0,\dots,N, k = 1,\dots,n_u)$، مسئله بهینه‌سازی می‌تواند به‌صورت زیر فرمول‌بندی شود:

$$\begin{aligned}
\min_{w_i,v_i} J_T = \min_{w_i,v_i} & \sum_{i=1}^{N-1} q'w_i + \sum_{i=0}^{N-1} r'v_i + p'w_N \\
& \text{با توجه به} \\
s_{ij} &\le w_{ij} \\
-s_{ij} &\le w_{ij} \\
u_{ik} &\le v_{ik} \\
-u_{ik} &\le v_{ik} \\
\mathbf{s}_{i+1} &= \mathbf{A}_{\mathbf{s}_i} + \mathbf{B}_{\mathbf{u}_i}
\end{aligned} \tag{۷.۲}$$

که اندیس j و k به ترتیب بر مولفه‌های حالت و بردار ورودی دلالت دارد. مسئله بهینه‌سازی یک برنامه خطی است که برای حل آن بسته‌های نرم‌افزاری کارآمد و موثر بهینه‌سازی وجود دارد.

۲.۲.۲ فرمول بندی برای چند ربات

برنامه خطی (۷.۲) می‌تواند به‌راحتی برای چند ربات گسترش یابد که هر ربات باید به مقصد مختلفی برود. به‌منظور تخصیص یک مقصد دلخواه، تابع هزینه (۶.۲) اصلاح‌شده است تا اختلاف بین مبدا و مقصد را کمینه کند. در این حالت ،بردار ورودی و بردارهای حالت مقصد ربات P^{th} به ترتیب به‌صورت $\mathbf{s}_{pf}$و$\mathbf{s}_{pi}, \mathbf{u}_{pi}$ هستند. در حالت کلی، تمام ربات‌ها می‌توانند ماتریس‌های سیستمی مختلف $(\mathbf{A}_p, \mathbf{B}_p)$ داشته باشند. علاوه بر این بردارهای وزن مختلف $\mathbf{q}_p$، $\mathbf{r}_p$ و $\mathbf{p}_p$ می‌تواند برای هر یک از ربات‌ها استفاده شوند. تابع هزینه برای ربات $\mathbf{p}^{th}$ به‌صورت زیر اصلاح‌شده است:

$$J_{T,p} = \sum_{i=1}^{N-1} q'_p|s_{pi} - s_{pf}| + \sum_{i=0}^{N-1} r'_p|u_{pi}| + p'_p|s_{pN} - s_{pf}| \tag{۸.۲}$$

تابع هزینه کلی برای K ربات می‌تواند به‌صورت مجموع توابع هزینه‌ی جدا از هم تعریف شود. بنابراین برنامه خطی به‌صورت زیر تعریف می‌شود:

$$\begin{aligned}
\min_{\mathbf{w}_{pi},\mathbf{v}_{pi}} \sum_{p=1}^{K} \Big(\sum_{i=1}^{N-1} \mathbf{q}'_p \mathbf{w}_{pi} + \sum_{i=0}^{N-1} \mathbf{r}'_p \mathbf{v}_{pi} + \mathbf{p}'_p \mathbf{w}_{pN} \Big) \\
s.t. \qquad s_{pij} - s_{pf} &\le w_{pij} \\
-s_{pij} + s_{pf} &\le w_{pij} \\
u_{pik} &\le v_{pik} \\
-u_{pik} &\le v_{pik} \\
\mathbf{s}_{p,i+1} &= \mathbf{A}_p \mathbf{s}_{pi} + \mathbf{B}_p \mathbf{u}_{pi}
\end{aligned} \tag{۹.۲}$$

در بخش بعدی ما محدودیت‌های اجتناب از موانع را معرفی می‌کنیم و سپس آن‌ها را به‌عنوان محدودیت‌های خطی/صحیح آمیخته روی حالت‌هایی از ربات‌ها فرمول‌بندی می‌کنیم.

۳.۲ محدودیت‌های دو دویی برای ممانعت از برخورد

۱.۳.۲ موانع ایستا

برای سادگی درک مسئله، یک ربات در حال حرکت را در فضای دوبعدی در نظر بگیرید که باید از برخورد آن با یک مانع مستطیلی جلوگیری شود. موقعیت مانع توسط مختصات پایین سمت چپ آن و بالا سمت راست محل یابی می‌شود: $(x_{\min}, y_{\min})$و$(x_{\max}, y_{\max})$. در هر گام زمانی i موقعیت (x_i, y_i) ربات باید در ناحیه‌ای خارج از مانع قرار بگیرد. این می‌تواند به‌صورت زیر فرمول نویسی شود:

$$\begin{aligned}
\forall i \in [1, \ldots, N]: \quad & x_i \le x_{\min} \\
\textbf{or} \quad & x_i \ge x_{\max} \\
\textbf{or} \quad & y_i \le y_{\min} \\
\textbf{or} \quad & y_i \ge y_{\max}
\end{aligned} \tag{۱۰.۲}$$

در اینجا فرض می‌کنیم که ربات، یک نقطه در حال حرکت است. موانع با توجه به‌اندازه شی متحرک، بزرگ شده‌اند [۲].

راهی برای تبدیل محدودیت‌های ـ یا (۱۰.۲) به محدودیت‌های ـ و این است که متغیرهای دودویی تعریف شوند [۷]. t_{ik} یک متغیر دودویی $(0,1)$ است و M یک عدد دلخواه بزرگ مثبت می‌باشد. بنابراین محدودیت‌های (۱۰.۲) می‌تواند توسط محدودیت‌های خطی/عدد صحیح

زیر جایگزین شود:

$$\begin{aligned} \forall i \in [۱, \ldots, N] : \quad & x_i \leq x_{\min} + Mt_{i۱} \\ \textbf{and} \quad & -x_i \leq -x_{\max} + Mt_{i۲} \\ \textbf{and} \quad & y_i \leq y_{\min} + Mt_{i۳} \\ \textbf{and} \quad & -y_i \leq -y_{\max} + Mt_{i۴} \\ \textbf{and} \quad & \sum_{k=۱}^{۴} t_{ik} \leq ۳ \end{aligned} \tag{۱۱.۲}$$

اگر محدودیت‌ـ یا اصلی K^{th} در گام زمانی i^{th} ارضا نشود، متغیر دو دویی t_{ik} مربوط، معادل ۱ است. آخرین محدودیت‌ـ و تضمین می‌کند که حداقل یکی از محدودیت‌های‌ـ یا ارضا شده است که اطمینان می‌دهد که از برخورد ربات به مستطیل جلوگیری می‌شود. محدودیت‌های (۱۱.۲) می‌تواند برای هر مانع در فضای مانور و برای هر رباتی در حالت چند رباتی فرمول‌بندی شود.

یک مانع با شکل دلخواه می‌تواند توسط چندضلعی اطراف آن توصیف شود که لبه‌ها محدودیت‌های ضد برخورد در هردو جهت x و y دارند. گسترش حرکت سه بعدی با موانع سه بعدی ساده است: یکی از احتیاجات برای فرمول بندی محدودیت‌های اضافی در جهت z و مانع سه بعدی می‌تواند توسط چند سطحی اطراف آن تعریف شود.

۲.۳.۲ موانع پویا

راهکار راحت برای این مسئله تعریف موانع جدید در هر گام زمانی است که به موقعیت موانع پویا در آن لحظه مرتبط است. مختصات موانع در هر گام زمانی، بر طبق حرکت از پیش تعیین‌شده آن‌ها، تغییر می‌کند . برای حرکت‌های انتقالی موانع مستطیلی، فرمت دودویی برای محدودیت‌های (۱۱.۲) همانند قبل باقی می‌ماند. از سوی دیگر حرکت چرخشی و حرکت موانع غیر مستطیلی محدودیت‌هایی در هر دو جهت x و y ایجاد می‌کنند.

۳.۳.۲ چند ربات

در حالتی که چند ربات در مسیرهاي مختلف حرکت کنند، اجتناب از برخورد بین آن‌ها با حالت موانع ایستا ارتباط دارد. در هر گام زمانی هر جفت از وسایل p و q باید فاصله کمینه بین هر یک در محور x و y داشته باشند. با توجه به برنامه حرکت ، موقعیت وسایل p و q در گام زمانی i^{th} به ترتیب به‌صورت (x_{qi}, y_{qi})و (x_{pi}, y_{pi}) است. فاصله ایمن توسط d_x و d_y نشان داده می‌شود. محدودیت‌ها را مي‌توان بصورت زیر توصیف کرد:

$$\begin{aligned} \forall i \in [۱, \ldots, N] : \forall p, q | q > p : \quad & |x_{pi} - x_{qi}| \geq d_x \\ \textbf{or} \quad & |y_{pi} - y_{qi}| \geq d_y \end{aligned} \tag{۱۲.۲}$$

شرط $q > p$ از دو بار تکرار شدن قیود در موقعیت‌ها ممانعت می‌کند. حذف کردن مقادیر مطلق محدودیت‌ها را به شکل زیر تبدیل می‌کند:

$$\begin{aligned} \forall i \in [۱, \ldots, N] : \forall p, q | q > p : \quad & x_{pi} - x_{qi} \geq d_x \\ \textbf{or} \quad & x_{qi} - x_{pi} \geq d_x \\ \textbf{or} \quad & y_{pi} - y_{qi} \geq d_y \\ \textbf{or} \quad & y_{qi} - y_{pi} \geq d_y \end{aligned} \tag{۱۳.۲}$$

این محدودیت‌ها می‌تواند به‌صورت محدودیت‌های عدد صحیح خطی با معرفی متغیر مناسب b_{pqik} فرمول‌بندی شود:

$$\begin{aligned} \forall i \in [۱, \ldots, N] : \quad \forall p, q | q > p : & \\ x_{pi} - x_{qi} \quad &\geq \quad d_x - M b_{pqi۱} \\ \textbf{and} \quad x_{qi} - x_{pi} \quad &\geq \quad d_x - M b_{pqi۲} \\ \textbf{and} \quad y_{pi} - y_{qi} \quad &\geq \quad d_y - M b_{pqi۳} \\ \textbf{and} \quad y_{qi} - y_{pi} \quad &\geq \quad d_y - M b_{pqi۴} \\ \textbf{and} \quad \sum_{k=۱}^{۴} b_{pqik} \quad &\leq \quad ۳ \end{aligned} \tag{۱۴.۲}$$

۴.۲ مدل برنامه‌ریزي رياضي

ترکیب برنامه خطی (۹.۲) با محدودیت‌های دودویی برای ممانعت از برخورد (۱۱.۲) و (۱۴.۲)، برنامه بزرگ غیر محدب ترکیبی عدد صحیح خطی آمیخته را می‌دهد. همچنین می‌توان محدودیت‌های خطی یا تکه‌ای‌ـخطی یعنی مقدار دقیق به‌عنوان بیشینه و کمینه ورودی و ایستگاه قرار داد [۸].

فرض کنید که برای هر وسیله ، بردارهای حالت و ورودی به صورت زیر باشند:

$$\mathbf{s}_p \quad = \quad [x_p, y_p, \dot{x}_p, \dot{y}_p]^T \tag{۱۵.۲}$$

$$\mathbf{u}_p \quad = \quad [u_{xp}, u_{yp}]^T \tag{۱۶.۲}$$

برای سادگی تنها موقعیت و سرعت به‌عنوان متغیرهای حالت در نظر گرفته شده‌اند. کمینه و بیشینه بردار ورودی و بردارهای حالت به‌صورت $s_{p,\min}$، $s_{p,\max}$، $u_{p,\min}$ و $u_{p,\max}$ هستند. محدودیت‌ها براي موقعیت ، باید به‌عنوان ناحیه‌ای مستطیلی بین مرزهایی که فرض شده است وسایل در آن بماند، تفسیر شود. فرض کنید K ربات و L مانع ایستا وجود دارند.

محدوده کامل زمانی به N گام زمانی تقسیم می‌شود. برنامه ترکیبی خطی عدد صحیح کامل به شکل زیر در می‌آید:

$$\min_{\mathbf{w}_{pi},\mathbf{v}_{pi}} \sum_{p=۱}^{k} \left(\sum_{i=۱}^{N-۱} \mathbf{q}'_p \mathbf{w}_{pi} + \sum_{i=۰}^{N-۱} \mathbf{r}'_p \mathbf{v}_{pi} + \mathbf{p}'_p \mathbf{w}_{pN} \right) \tag{۱۷.۲}$$

با توجه به $\forall p \in [۱,\ldots,K]:$

$$\begin{aligned}
\forall i \in [۱,\ldots,N]: \; x_{pi} - x_{pf} &\leq w_{pi۱} \\
-x_{pi} + x_{pf} &\leq w_{pi۱} \\
y_{pi} - y_{pf} &\leq w_{pi۲} \\
-y_{pi} + y_{pf} &\leq w_{pi۲} \\
\dot{x}_{pi} - \dot{x}_{pf} &\leq w_{pi۳} \\
-\dot{x}_{pi} + \dot{x}_{pf} &\leq w_{pi۳} \\
\dot{y}_{pi} - \dot{y}_{pf} &\leq w_{pi۴} \\
-\dot{y}_{pi} + \dot{y}_{pf} &\leq w_{pi۴}
\end{aligned} \tag{۱۸.۲}$$

$$\begin{aligned}
\forall i \in [۰,\ldots,N-۱]: \; u_{xpi} &\leq v_{pi۱} \\
- \; u_{xpi} &\leq v_{pi۱} \\
u_{ypi} &\leq v_{pi۲} \\
- \; u_{ypi} &\leq v_{pi۲}
\end{aligned} \tag{۱۹.۲}$$

$$\begin{aligned}
&\forall i \in [۰,\ldots,N-۱]: \\
\mathbf{s}_{p(i+۱)} &= \mathbf{A}_p \mathbf{s}_{pi} + \mathbf{B}_P \mathbf{u}_{pi} \\
\mathbf{s}_{po} &= [x_{po}, y_{po}, \dot{x}_{po}, \dot{y}_{po}]^T
\end{aligned} \tag{۲۰.۲}$$

$$\begin{aligned}
\forall i \in [۰,\ldots,N-۱]: \; \mathbf{u}_{pi} &\geq \mathbf{u}_{p,\min} \\
\mathbf{u}_{pi} &\leq \mathbf{u}_{p,\max}
\end{aligned} \tag{۲۱.۲}$$

$$\begin{aligned}
\forall i \in [۱,\ldots,N]: \; \mathbf{s}_{pi} &\geq \mathbf{s}_{p,\min} \\
\mathbf{s}_{pi} &\leq \mathbf{s}_{p,\max}
\end{aligned} \tag{۲۲.۲}$$

$$
\begin{aligned}
&\forall c \in [۱,\ldots,L], i \in [۱,\ldots,N]: \\
&\quad x_{pi} \leq x_{c,\min} + Mt_{pci۱} \\
&\quad -x_{pi} \leq -x_{c,\max} + Mt_{pci۲} \\
&\quad y_{pi} \leq y_{c,\min} + Mt_{pci۳} \\
&\quad -y_{pi} \leq -y_{c,\min} + Mt_{pci۴} \\
&\quad \sum_{k=۱}^{۴} t_{pcik} \leq ۳ \\
&\quad t_{pcik} \in \{۰,۱\}
\end{aligned}
\tag{۲۳.۲}
$$

$$
\begin{aligned}
&\forall i \in [۱,\ldots,N] : q \in [p+۱,\ldots,K]: \\
&\quad x_{pi} - x_{qi} \geq d_x - Mb_{pqi۱} \\
&\quad x_{qi} - x_{pi} \geq d_x - Mb_{pqi۲} \\
&\quad y_{pi} - y_{qi} \geq d_y - Mb_{pqi۳} \\
&\quad y_{qi} - y_{pi} \geq d_y - Mb_{pqi۴} \\
&\quad \sum_{k=۱}^{۴} b_{pqik} \leq ۳ \\
&\quad b_{pqik} \in \{۰,۱\}
\end{aligned}
\tag{۲۴.۲}
$$

مسئله‌اي كه مطرح كرديم يك مسئله سخت است كه وقتي مسئله بزرگ مي‌شود حل آن زمان‌بر و حتي غيرممكن است . به همين دليل همان‌طور كه در فصل‌هاي بعد خواهيم ديد ، براي حل اين مسئله به سراغ روش‌هاي فراابتكاري مي‌رويم كه بتوانيم در زمان قابل قبول به يك جواب بهينه يا نزديك به بهينه برسيم.

فصل ۳

طراحي كنترل‌كننده فازي براي مسيريابي ربات‌ها

۱.۳ مقدمه

در یک محیط دارای مانع، مسیریابی شامل یافتن یک مسیر مناسب و بدون برخورد از مبدا به مقصد است؛ بنابراین ربات‌ها باید چندین مسئله مختلف مانند اجتناب از مانع، شناسایی موقعیت، موارد مربوط به ایمنی و غیره را باید حل کنند که پیش‌نیاز طراحي یک الگوریتم هوشمندانه برای تعیین مسیر مناسب به سمت موقعیت هدف است.

در سال‌های اخیر، پژوهشگران بسیاری مسئله برنامه‌ریزی مسیر را مورد کنکاش قرار داده‌اند و روش‌های گوناگونی مانند شبکه‌های عصبی، الگوریتم ژنتیک [۱۱] بینایی مصنوعی و کنترل PID [۱۳] برای حل این مسئله در محیط‌های ایستا پیشنهاد شده است. این روش‌ها معمولا در برنامه‌ریزی مسیر سراسری استفاده می‌شود و کنترل بلادرنگ (که سیستم باید به شرایط جدید طی زمان مشخصی واکنش نشان دهد) نمی‌توان از آن بهره جست. البته برخی از روش‌ها شامل کنترل بلادرنگ نیز برای ناوبری در محیط‌های ایستا توسعه‌یافته‌اند. به‌عنوان نمونه در [۱۴] یک روش مبتنی بر یک الگوریتم کنترل از راه دور پیشرفته برای تعیین

مسیریاب پیشنهاد شده است که اطلاعات حسگرها را با اطلاعات دیداری فرد کنترل‌کننده ترکیب می‌کند. در مرجع [۱۵] پژوهشگران روشی به‌نام دایره هدایت گسترش‌یافته برای ناوبری ربات‌ها و بهبود کارایی عملیات از راه دور توسعه داده‌اند.

تعدادی از روش‌های قدیمی طراحی‌شده برای مسیریابی در محیط‌های ایستا نیز برای مسیریابی در محیط‌های پویا بسط داده شده‌اند؛ مانند روش میدان بالقوه [۱۶] ، روش نقشه راه‌ها [۱۷] و روش پنجره غلتان [۱۸]. این روش‌ها به خوبی در محیط ایستا می‌توانند اجرا شوند اما برخی از آن‌ها در محیط پویا دچار مشکل می‌شوند؛ زیرا یکی از مفروضات این روش‌ها، ایستا بودن موانع در بازه‌های زمانی معین است.

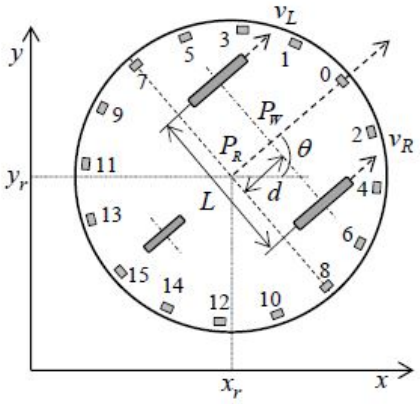
مسئله مسیریابی برای دستگاه‌های چند رباتی نیز از دیگر زمینه‌های مورد استقبال پژوهشگران است. تیاگو در [۱۹] مسئله کنترل چند رباتی را با استفاده از نوعی ماشین فازی تاکاگی‌ـ سوگنو مورد مطالعه قرار داده و ژانگ در [۲۰] روش جدید شتاب‌ـ تغییرـ فاصله را به‌وسیله استفاده از تغییرات سرعت و شتاب برای ربات‌ها برای برنامه‌ریزی حرکت در محیط‌های پویا معرفی کرده است. البته در این روش باید اندازه، موقعیت و شتاب همه موانع و ربات‌ها به‌صورت برخط اندازه‌گیری شود. دایال در [۲۱] مسئله تعیین مسیر برای ربات‌های متحرک را به‌وسیله استفاده از روش فازی‌ـ عصبی مطالعه کرده که در این روش نیز ارتباط درونی برخط بین ربات‌ها لازم است؛ بنابراین به نظرمی‌آید مطالعه مسئله پیشنهادی در محیطی کاملا ناشناخته ضرورت دارد. در این فصل با استفاده از منطق فازی مسئله مسیریابی ربات‌ها در محیط‌های پویای ناشناخته را حل خواهیم‌کرد.

در این راستا با استفاده از ضریب خطر که به‌وسیله اطلاعات حسگرها و استنتاج فازی تخمین زده می‌شود، ربات‌ها از موانع ایستا و پویا اجتناب می‌کنند و سپس به‌وسیله یک کنترل‌کننده فازی دیگر به جستجوی هدف می‌پردازند. به‌علاوه فرض به این است که همه ربات‌ها شبیه یکدیگر هستند و ازالگوریتم مسیریابی یکسان استفاده می‌کنند.

۲.۳ مدل‌سازی ربات

در این بخش، ما یک ربات چرخ‌دار متحرک کلاسیک را به عنوان مثال برای شبیه‌سازی انتخاب می‌کنیم. دو موتور DC به‌ترتیب بر روی چرخ‌های چپ و راست نصب شده‌اند و هر طرف دارای یک رمزگذار روی آن هستند و ربات می‌تواند تشخیص دهد که هر چرخ چه مسافتي را طی کرده‌است. معمولا در عمل یک حسگر چرخشی بر روی بدنه ربات برای اطلاعات بازخورد و اصلاح مسیر وجود دارد.

ساختار چنین رباتی دایره‌ای شکل می‌باشد که در شکل ۱.۳ نشان داده شده است. درمجموع ۱۶ حسگر مجاورت (حسگرهای فراصوتی یا حسگرهای مادون‌قرمز) به‌طور ثابت در اطراف ربات با فاصله یکسان نصب شده است. این حسگرها که از s_0 تا s_{15} شماره‌گذاری می‌شوند، برای پیدا کردن فاصله موانع اطراف با سایر ربات‌ها استفاده می‌شود.



شکل ۱.۳: مدل‌سازی ربات چرخ متحرک

در این‌جا حالت‌هایی را فرض می‌کنیم که همه موانع و دیگر ربات‌ها برای ربات ناشناخته هستند. داده‌هایی که یک عنصر مسیر را تعریف می‌کنند می‌توانند به‌عنوان مجموعه‌ای از مقادیر مختصات تعیین نقاط مشخص شوند. فرض کنید P_w و P_R نقاط مرکزی بین دو چرخ و مرکز ربات باشند. ما از مختصات تعمیم‌یافته $[x, y, \theta]^T$ برای توصیف وضعیت هر نقطه استفاده می‌کنیم؛ بنابراین، توابع حرکتی از P_w می‌توانند به‌صورت زیر باشند:

$$\begin{cases} \dot{x} = \frac{v_L + v_R}{۲} \cos\theta \\ \dot{y} = \frac{v_L + v_R}{۲} \sin\theta \\ \dot{\theta} = \frac{v_L - v_R}{L} \end{cases} \tag{۱.۳}$$

جایی که v_R و v_L سرعت خطی متناظر با چرخ‌های سمت راست و چپ است. θ نشان‌دهنده زاویه بین جهت رانندگی ربات و محور x و L به‌عنوان فاصله بین دو چرخ اندازه‌گیری می‌شود. فرض می‌کنیم که ربات تحت شرایطی بدون هیچ لغزشی حرکت می‌کند، بنابراین ربات باید محدودیت غیرخطی زیر را دنبال کند:

$$x \sin\theta - y \cos\theta = ۰ \tag{۲.۳}$$

فاصله d به‌عنوان طول بین P_w و P_R است. با توجه به روش هندسی، عملکرد حرکتی P_R می‌تواند به‌راحتی از (۱.۳) به‌صورت زیر تغییر کند:

$$\begin{cases} x = \frac{v_L + v_R}{۲} \cos\theta - d\cos\theta \\ y = \frac{v - L + v_R}{۲} \sin\theta - d\sin\theta \\ \theta = \frac{v_L - v_R}{L} \end{cases} \tag{۳.۳}$$

۳.۳ طراحی کنترل‌کننده منطق فازی

هدف ما هدایت ربات‌های متحرک با یک مسیر بدون برخورد است تا در یک محیط پویا برای رسیدن به موقعیت هدف به‌طور موفقیت‌آمیز حرکت کند. مسئله برنامه‌ریزی مسیر را می‌توان به‌عنوان اجتناب از مانع و جهت‌یابی هدف درنظرگرفت . بنابراین در این فصل دو نوع کنترل‌کننده با اثربخشی متفاوت توسعه خواهد یافت.

۱.۳.۳ کنترل‌کننده اجتناب از مانع

در موقعیت اولیه، ربات رو به جلو در جهت هدف حرکت خواهد کرد. هرگاه یکی ازحسگرهای نصب شده بر روی بدنه ربات، جسمی (موانع یا ربات‌های دیگر) را شناسایی کند، یک راهبرد اجتناب فعال می‌شود. این راهبرد با استفاده از یک کنترل‌کننده منطق فازی به نام OA-FLC مبتنی بر مجموعه‌ای از قوانین به‌دست آمده است. در این فصل، ما از بردار سرعت v برای پیدا کردن سرعت ربات‌ها و موانع متحرک استفاده می‌کنیم، یعنی $\vec{v} = [v, \theta]^T$. همان‌طور که در شکل ۲.۳ نشان داده شده است، P_R و P_O نشان دهنده موقعیت فعلی یک ربات و یک مانع است و به‌ترتیب با سرعت جریان $\vec{v_r}$ و $\vec{v_o}$ می‌باشد. سرعت نسبی ربات نسبت به مانع $\vec{v_{r_O}} = \vec{v_r} - \vec{v_o}$ است. در این شکل، جهت بردار $\vec{v_{r_O}}$ به سمت مانع است. این بدان معنی است که اگر چیزی در حال حاضر انجام نشود، برخوردی بین ربات و مانع به وجود خواهد آمد. حسگرهای نصب شده بر روی بدنه ربات برای هر ۰.۱ ثانیه یک بار اجرا می‌شوند. بنابراین، در چنین زمان کوتاهی، می‌توانیم محاسبات سرعت نسبی را به‌صورت معادله زیر ساده کنیم تا حل تقریبی قابل قبول را به‌دست آوریم.

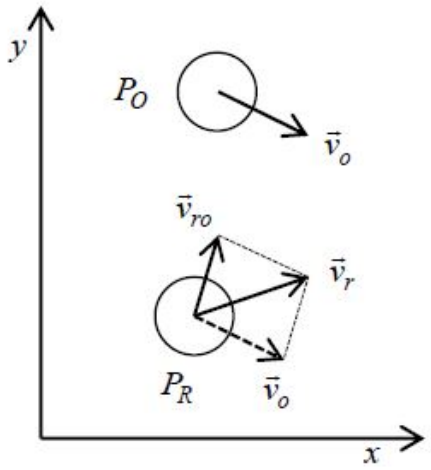
$$v^i_{ro} = \frac{d^i_{t_1} - d^i_{t_2}}{\Delta t}, \tag{۴.۳}$$

که $d^i_{t_1}$ و $d^i_{t_2}$ در فاصله مشخص شده به‌وسیله حسگرهای i $(i = 0, \ldots, 15)$ ام در زمان t_1 و t_2 تشخیص داده شده است. و فاصله زمانی Δt از t_1 و t_2، ۰.۱ ثانیه است. از زمان مورد انتظار در برخورد برای نشان‌دادن ضریب خطر استفاده می‌کنیم:

$$R^i_{dc} = \frac{d^i_{t_2}}{v^i_{ro}}. \tag{۵.۳}$$

هرچه مقدار R^i_{dc} کوچک‌تر باشد، ربات (مانع متحرك) خطرناک‌تر خواهد بود. به‌طور خاص، هنگامی‌که جهت بردار $\vec{v^i_{ro}}$ نقطه دور از مانع است، مقدار R^i_{dc} به‌صورت یک مجموعه نامتناهی تعیین خواهد شد. در این‌جا مینیمم ضریب خطر و زاویه حسگر مربوطه به‌عنوان متغیرهای ورودی OA-FLC مورداستفاده قرارمی‌گیرند. علاوه بر این، سرعت خطی چرخ‌های چپ و راست متغیرهای خروجی خواهد بود. می‌توانیم این را به‌صورت زیر نشان دهیم:

$$\mathbb{R}_{dc} = \{R^i_{dc} | 0 \leq i \leq 15\}. \tag{۶.۳}$$



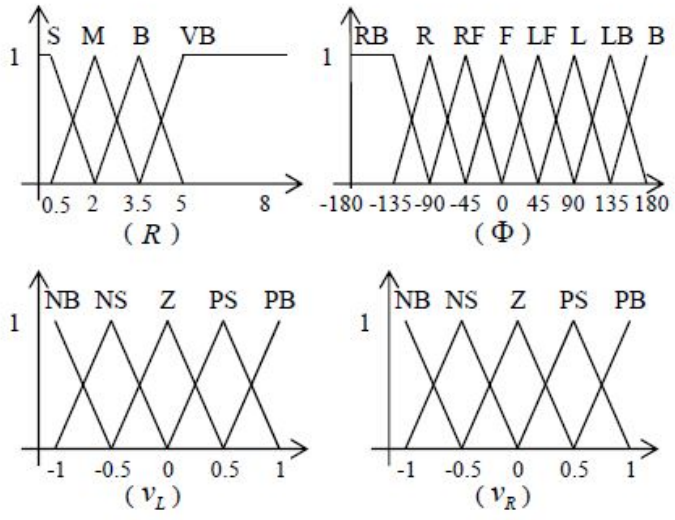
شکل ۲.۳: ساختار سرعت نسبی

بنابراین متغیرهای ورودی OA-FLC را می‌توان به‌صورت زیر توصیف کرد:

$$R = \min(\mathbb{R}_{dc}) \tag{۷.۳}$$

$$\phi = \angle s_i, \qquad R^i_{dc} = \min(\mathbb{R}_{dc}) \tag{۸.۳}$$

که $(-۱۸۰^o \leq \angle s_i < ۱۸۰^o)$ زاویه‌ی مربوط به حسگر i ام است.



شکل ۳.۳: توابع عضویت برای متغیرهای ورودی و خروجی OA-FLC

فرآیند فازی سازی یک تبدیل FLC یک مقدار ورودی غیر فازی را به یک مقدار فازی تبدیل می‌کند. برای فازی سازی ، متغیرهای ورودی و خروجی را می‌توان به چندین اصطلاح زبانی تقسیم کرد:

s: کوچک M: متوسط B: بزرگ VB: بسیار بزرگ
RB: راست عقب R: راست RF: راست جلو F: جلو
LF: جلو چپ L: چپ LB: چپ عقب B: عقب
NB: منفی بزرگ NS: منفی کوچک Z: صفر
PB: مثبت بزرگ PS: مثبت کوچک

توابع عضویت برای متغیرهای ورودی و خروجی OA-FLC همان طور که در شکل ۳.۳ نشان داده‌شده است ، هنگامی که خطر اصلی از سمت چپ باشد زاویه Φ مثبت است، در غیراین‌صورت منفی است. علاوه‌براین، از توابع عضویت از نوع مثلثی در محدوده $[-۱,۱]$ برای متغیرهای خروجی استفاده می‌کنیم که نشان‌دهنده مقیاس حداکثر سرعت خطی است.
مجموعه قوانینی که برای OA-FLC اعمال می‌شود، در جدول ۱.۳ نشان داده‌شده است.

جدول ۱.۳: قوانین برای OA-FLC

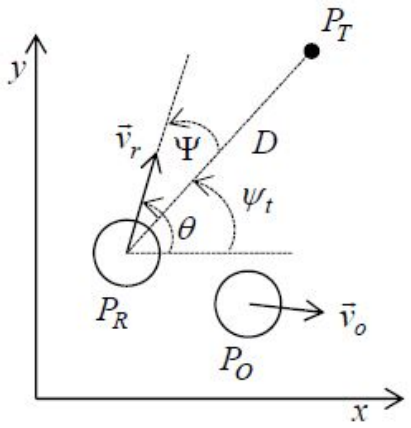
	V_L	Φ							
		RB	**R**	**RF**	**F**	**LF**	**L**	**LB**	**B**
R	**S**	PB	PB	NB	NB	NS	PB	PB	PS
	M	PB	PB	NB	NB	PB	PB	PS	PS
	B	PB	PB	NS	Z	PB	PB	PS	PS
	VB	PB	PB	PS	PS	PB	PB	PS	PS

	V_R	Φ							
		RB	**R**	**RF**	**F**	**LF**	**L**	**LB**	**B**
R	**S**	PB	PB	NS	NS	NB	PB	PB	PS
	M	PS	PB	PB	Z	NB	PB	PB	PB
	B	PS	PB	PB	PB	NS	PB	PB	PB
	VB	PB	PB	PB	PB	PS	PB	PB	PB

۲.۳.۳ کنترل‌کننده جهت‌یابی هدف

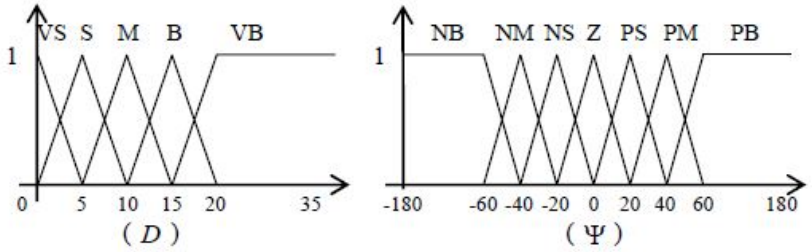
هنگامی که خطر برخورد بین ربات و موانع دیگر شامل موانع ایستا و پویا وجود ندارد یا ربات به‌طور موقت روی موانع رانده شده است، راهبرد دیگری برای جهت‌گیری هدف فعال خواهد شد. این راهبرد توسط یک کنترل‌کننده فازی دیگر به‌نام TO-FLC به دست می‌آید.

مدل شماتیک جهت‌گیری هدف در شکل ۴.۳ آمده است. در این شکل، با موقعیت P_R فعلی ربات پیش از مانع P_O حرکت کرده است. با زاویه Ψ_t بین خط از مرکز ربات تا نقطه هدف و محور x ، نشان می‌دهیم که اختلاف زاویه‌ای $\Psi = \theta - \psi_t$ است. در حال حاضر، مانع درحال دور شدن از ربات است و هیچ خطری برای ربات ندارد. بنابراین راهبرد TO-FLC به کار می‌رود. این کنترل کننده برای حرکت دادن ربات به‌سوی موقعیت هدف استفاده شده است. به‌عبارت دیگر، از آن برای کاهش اختلاف زاویه Ψ که یکی از متغیرهای ورودی TO-FLC است استفاده می‌شود.



شکل ۴.۳: طرح جهت‌گیری هدف

یکی دیگر از محدودیت‌های TO-FLC ، فاصله D بین ربات و هدف است . علاوه بر این، متغیرهای خروجی TO-FLC ، سرعت خطی چرخ‌های چپ و راست هستند که با متغیرهاي خروجي OA-FLC مشابه هستند. توابع عضویت برای متغیرهای ورودی TO-FLC و قوانین مربوطه در شکل ۵.۳ و جدول ۲.۳ نشان داده‌شده است.



شکل ۵.۳: توابع عضویت برای متغیرهای ورودی TO-FLC

جدول ۲.۳: قوانین برای TO-FLC

	V_L	Ψ						
		NB	**NM**	**NS**	**Z**	**PS**	**PM**	**PB**
D	**VS**	NB	NB	NS	PS	PS	PB	PB
	S	NB	NS	Z	PB	PB	PB	PB
	M	NB	NS	Z	PB	PB	PB	PB
	B	NB	NB	NS	PB	PB	PB	PB
	VB	NB	NB	NB	PB	PB	PB	PB

	V_R	Ψ						
		NB	**NM**	**NS**	**Z**	**PS**	**PM**	**PB**
D	**VS**	PB	PB	PS	PS	NS	NB	NB
	S	PB	PB	PB	PB	Z	NS	NB
	M	PB	PB	PB	PB	Z	NS	NB
	B	PB	PB	PB	PB	NS	NB	NB
	VB	PB	PB	PB	PB	NB	NB	NB

۴.۳ شرایط خاص در اجتناب از موانع

در محیط با موانع پویاي چندگانه، زمانی که مقدار زمان مورد انتظار برخورد کمتر از ”متوسط ” است، اگر موانع متعددی در مقابل ربات وجود داشته باشد و خطر اصلی مستقیما درهمان زمان رخ دهد یا اگر چندین مانع به طرف ربات حرکت کنند، در چنین شرایط پیچیده‌ای فقط استفاده از قوانین ”اگر ...آنگاه...” کافی نیست. این بخش وضعیت خاصی را موردبحث قرار می‌دهد؛ زمانی که خطر اصلی از سمت جلو، پشت، مستقیماً از چپ یا راست باشد. هنگامی‌که مقدار زمان مورد انتظار برخورد بزرگ‌تر از ”متوسط ” است، روش توسعه یافته در این بخش استفاده نخواهد شد.

ما از زاویه $\phi = \angle(v\vec{r}_O - \vec{v}_r)$ برای نشان دادن جهت خطر در پیش رو استفاده می‌کنیم. زاویه $\angle s_i$ نشان‌دهنده زاویه حسگر i ام است. بنابراین شرایط خاص را می‌توان به‌صورت زیر تعریف کرد:

از جلو: اگر $\phi \subset \{\angle s_0.\angle s_1, \angle s_2\}$ باشد.

از عقب: اگر $\phi \subset \{\angle s_{13}.\angle s_{14}, \angle s_{15}\}$ باشد.

به‌طور مستقیم از چپ: اگر $\phi \subset \{\angle s_5.\angle s_7, \angle s_9\}$ باشد.

به‌طور مستقیم از راست: اگر $\phi \subset \{\angle s_6.\angle s_8, \angle s_{10}\}$ باشد.

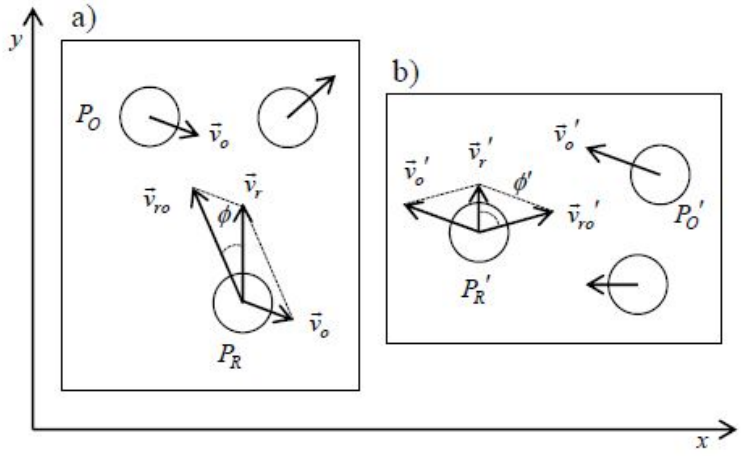
دو مورد از چهار موقعیت خاص در شکل ۶.۳ نشان داده شده است. در ارتباط با وضعیت (a) ، تفاوت فاصله بین ربات و مانع را در نظر می‌گیریم تا سمت چپ به جلو ربات و همچنین فاصله سمت راست به جلو موانع تعیین شود. تفاوت فاصله چپ و راست به جلو به‌صورت زیر تعریف می‌شود:

$$D^1_{dif} = \frac{s_1 + s_3}{2} - \frac{s_2 + s_4}{2}. \qquad (۹.۳)$$

بنابراین محدودیت را می‌توان چنین توضیح داد: اگر $D^1_{dif} > 0$ ، ربات به سمت چپ بچرخد، در غیر این‌صورت به سمت راست بچرخد . در مورد حالت (b) در شکل ۶.۳، خطر اصلی از سمت راست می‌آید. حرکت به جلو یا عقب تبدیل به یک مشکل مهم شده است. به‌طور مشابه، ما نشان می‌دهیم که چنین تفاوت فاصله‌ای از سمت راست به جلو و از سمت راست به عقب به‌صورت زیر است:

$$D^2_{dif} = \frac{s_4 + s_6}{2} - \frac{s_{10} + s_{12}}{2} \qquad (۱۰.۳)$$

به همین ترتیب، محدودیت را می‌توان توصیف کرد: اگر $D^2_{dif} > 0$ ربات به جلو حرکت می‌کند، در غیر این صورت به عقب حرکت می‌کند. به همان شیوه، روش مورد بحث فوق را می‌توان به‌راحتی به موقعیت مستقیما از عقب و مستقیما از چپ گسترش داد. با توجه به روش مورد بحث فوق و کنترل‌کننده‌های پیشنهادی طراحی شده در بخش ۳.۳، فرآیند حرکت برای برنامه‌ریزی مسیر می‌تواند به‌عنوان الگوریتم ۷.۳ توصیف شود.



شکل ۶.۳: شرایط ویژه زمانی که خطر اصلی از راست به جلو (a) و به‌طور مستقیم راست (b)

حسگر شماره i : s_i حد آستانه فاصله امن : D_{safe} حد آستانه زمان امن : T_{safe}

مختصات ربات : $[x_r,y_r]$ زاویه انحراف ربات از مقصد : Ψ فاصله تا مقصد : D

مجموعه ضرایب خطر : $\mathbb{R}_{dc}$ زاویه جهت‌های حرکت ربات و مانع : Φ

۱- مقداردهی اولیه : برای هر حسگر مقادیر D_{safe} و T_{safe} را تنظیم کن.

۲- استراتژی حرکت (کارهای زیر را تا زمان رسیدن ربات به مقصد تکرار کن) :

۲-۱- $R \leftarrow \min(\mathbb{R}_{dc})$

۲-۲- در صورتی که برای حسگر s_i داشته باشیم $R_{dc}^{i} = \min(\mathbb{R}_{dc})$ آنگاه

۲-۲-۱- $\Phi \leftarrow \angle s_i$

۲-۳- اگر فاصله مانع تا نزدیک‌ترین حسگر کمتر از D_{safe} است، آنگاه

۲-۳-۱- اگر $R < T_{safe}$ آنگاه

۲-۳-۱-۱- اگر $R <$ "medium" و شرایط خاص برقرار بود، آنگاه

۲-۳-۱-۱-۱- محاسبه مقدار Φ در شرایط خاص

۲-۳-۱-۲- کنترل به *OA-FLC* سپرده شود.

۲-۳-۲- در غیر اینصورت

۲-۳-۲-۱- مقادیر Ψ و D را محاسبه کن.

۲-۳-۲-۲- کنترل به *TO-FLC* سپرده شود.

۲-۴- در غیر اینصورت

۲-۴-۱- مقادیر Ψ و D را محاسبه کن.

۲-۴-۲- کنترل به *TO-FLC* سپرده شود.

۲-۵- مقدار $[x_r,y_r]$ را به‌روز رسانی کن

۳- پایان.

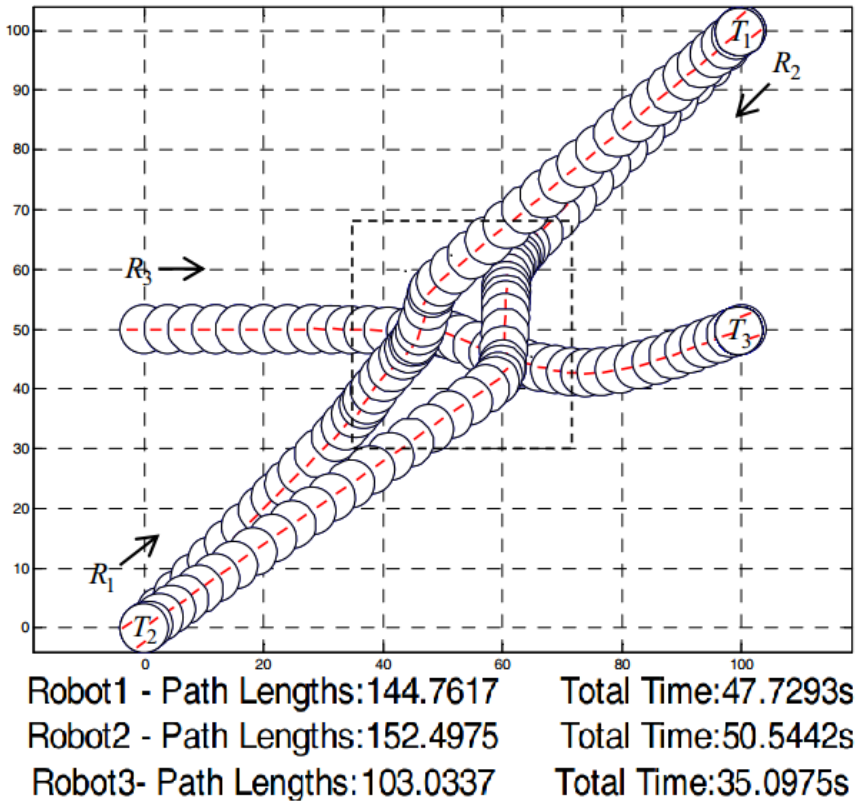
شکل ۷.۳: فرآیند حرکت مسیر برای برنامه‌ریزی مسیر

۵.۳ نتایج شبیه‌سازی

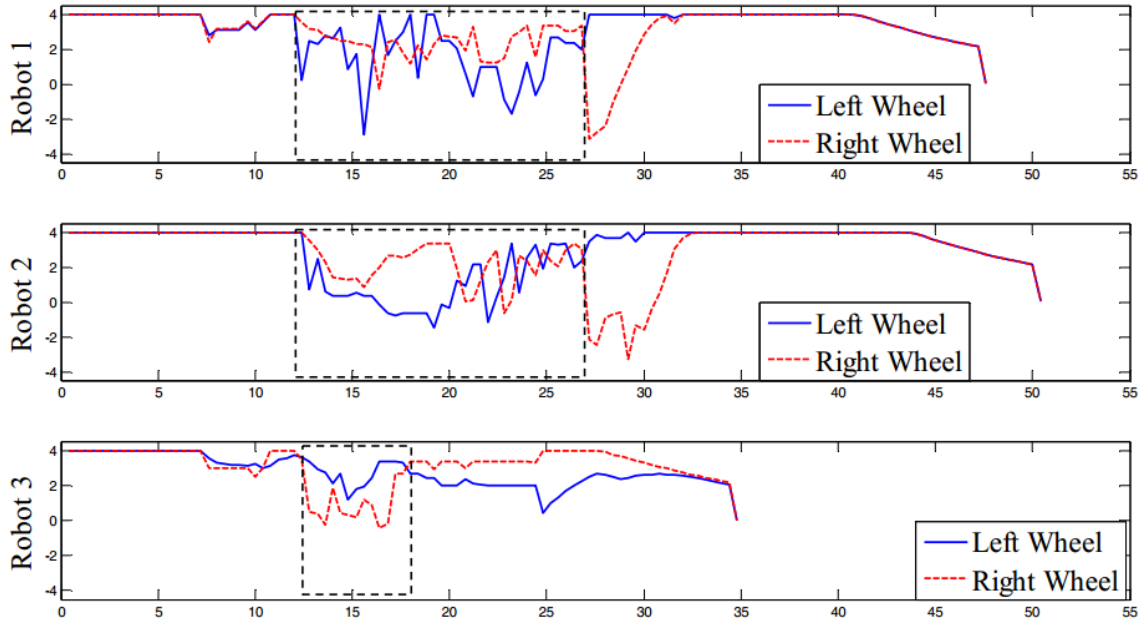
برای بررسی اثربخشی الگوریتم پیشنهادی، یک سری از شبیه‌سازی‌ها با مجموعه‌ای از کدهای Matlab برای محاسبات انجام‌شده است. ما از R_i، T_i برای نشان دادن ربات i و موقعیت هدف متناظر با آن استفاده می‌کنیم. O_i و M_i به ترتیب برای نشان دادن موانع پویا و ایستا i ام استفاده می‌شوند. تمام ربات‌ها در یک صفحه افقی با اندازه ۱۰۰ در ۱۰۰ حرکت می‌کنند. تمام ربات‌ها دارای اندازه یکسان با شعاع ۴ واحد هستند. هر ربات از همان راهبرد حرکت استفاده می‌کند و حداکثر سرعت چرخ‌های چپ و راست ۴ واحد در هر ثانیه است. موانع ایستا و پویا در این بخش شبیه‌سازی می‌شوند. علاوه‌براین، ما موقعیت ربات‌هایی که هر یک ثانیه گراف مسیر را تولید می‌کنند، ثبت خواهیم کرد.

۱.۵.۳ شبیه‌سازی در محیط بدون مانع

نتیجه‌ای از گراف مسیر در محیط بدون مانع در شکل ۸.۳ نشان داده‌شده است. سه ربات که به‌ترتیب از نقاط شروع [۰,۰]، [۱۰۰,۱۰۰]، [۰,۵۰] به نقاط هدف [۱۰۰,۱۰۰]، [۰,۰] و [۱۰۰,۵۰] حرکت می‌کنند وجود دارد. تمام ربات‌ها با حداکثر سرعت رو به جلو حرکت می‌کنند تازمانی که به منطقه خطرناک (جعبه سیاه نقطه‌چین) وارد شوند. سپس راهبرد OA-FLC فعال شده است. شکل ۹.۳ منحنی تغییر سرعت چرخ‌های چپ و راست برای ربات‌ها را نشان می‌دهد. ناحیه‌ها در جعبه سیاه نقطه‌چین نشان‌دهنده تغییرات سرعت در طی فرآیند اجتناب از مانع است. به‌عبارت دیگر، هنگامی‌که ربات متوجه شد که یک شی دیگر درحال نزدیک شدن است، سرعت را بلافاصله کاهش می‌دهد و با استفاده از استراتژی OA-FLC مسیر را به سمت چپ یا راست تنظیم می‌کند. پس از خروج از منطقه خطرناک، سرعت چرخ‌های چپ و راست به سمت نقطه هدف تنظیم می‌شود و دوباره به حداکثر سرعت می‌رسد.



شکل ۸.۳: نتیجه در محیط بدون مانع

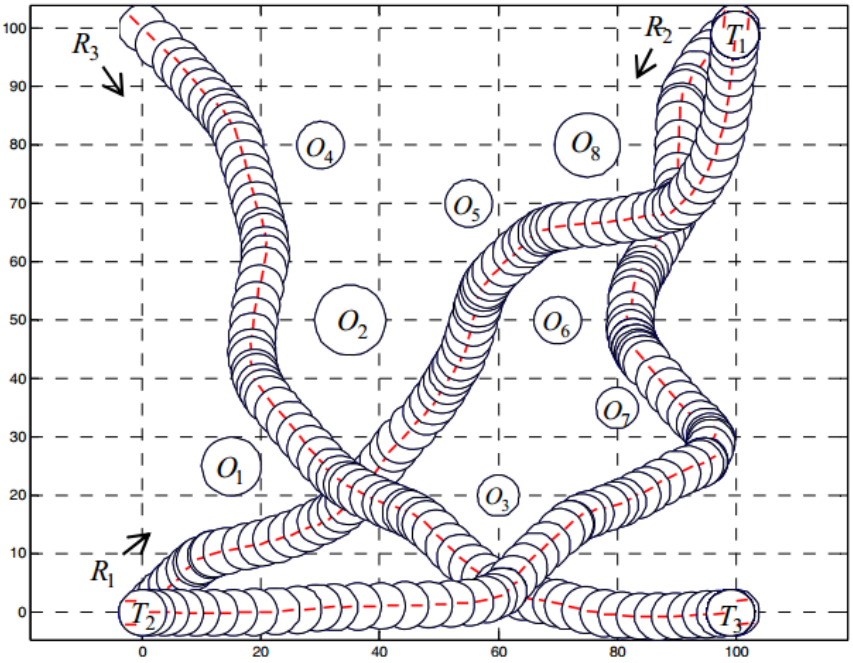


شکل ۹.۳: منحنی سرعت تغییر چرخ‌های چپ و راست در محیط بدون مانع

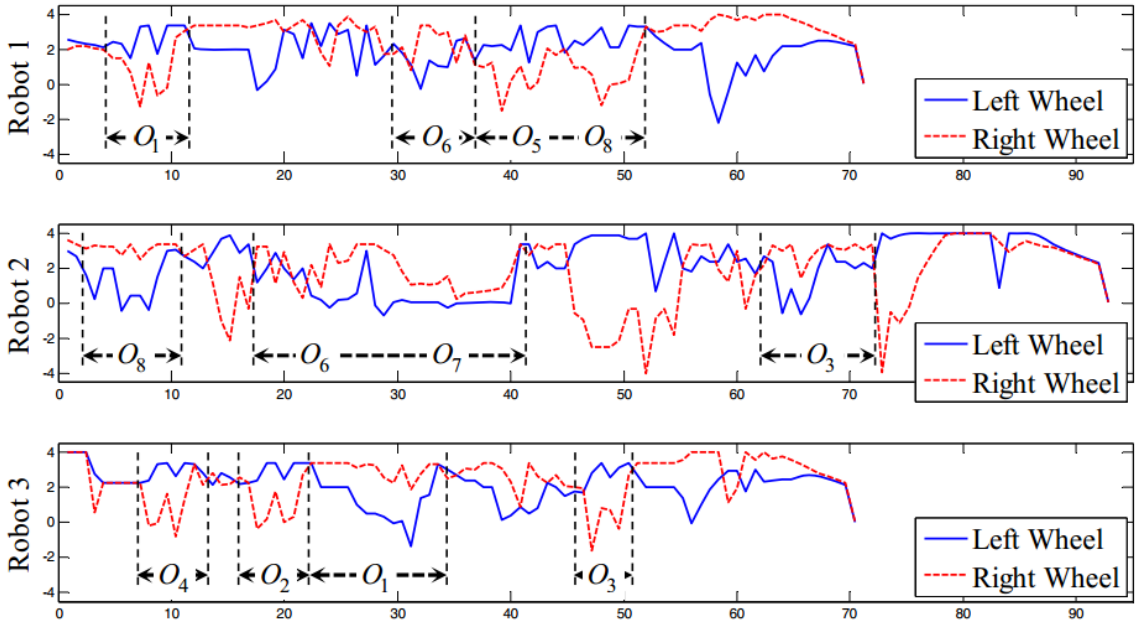
۲.۵.۳ شبیه‌سازی در محیط با موانع ایستا

نقاط شروع و هدف برای همه ربات‌ها و گراف مسیر نهایی در شکل ۱۰.۳ نشان داده شده است. تمام موانع در این بخش به‌صورت دایره‌ای گوشه‌دار و دارای اندازه متفاوت شعاع هستند. از این شکل می‌توان مشاهده کرد که الگوریتم پیشنهادی دارای عملکرد خوبی در محیط با موانع ایستا است.

منحنی تغییر سرعت چرخ‌های چپ و راست برای همه ربات‌ها در شکل ۱۱.۳ نشان داده‌شده است. ناحیه‌ها زمانی که ربات‌ها سعی در اجتناب از موانع که از O_1 تا O_8 شماره گذاری شده‌اند نیز در این شکل مشخص شده‌اند. در این مناطق مشخص شده، ربات‌ها مسیرهای صحیح را از طریق تنظیم اختلاف سرعت چرخ‌های چپ و راست به سمت چپ یا راست بر اساس محیط اطراف توسط راهبرد OA-FLC پیدا می‌کنند.



شکل ۳.۱۰: نتیجه در محیط موانع ایستا



شکل ۳.۱۱: منحنی سرعت تغییر چرخ‌های چپ و راست در محیط با موانع ایستا

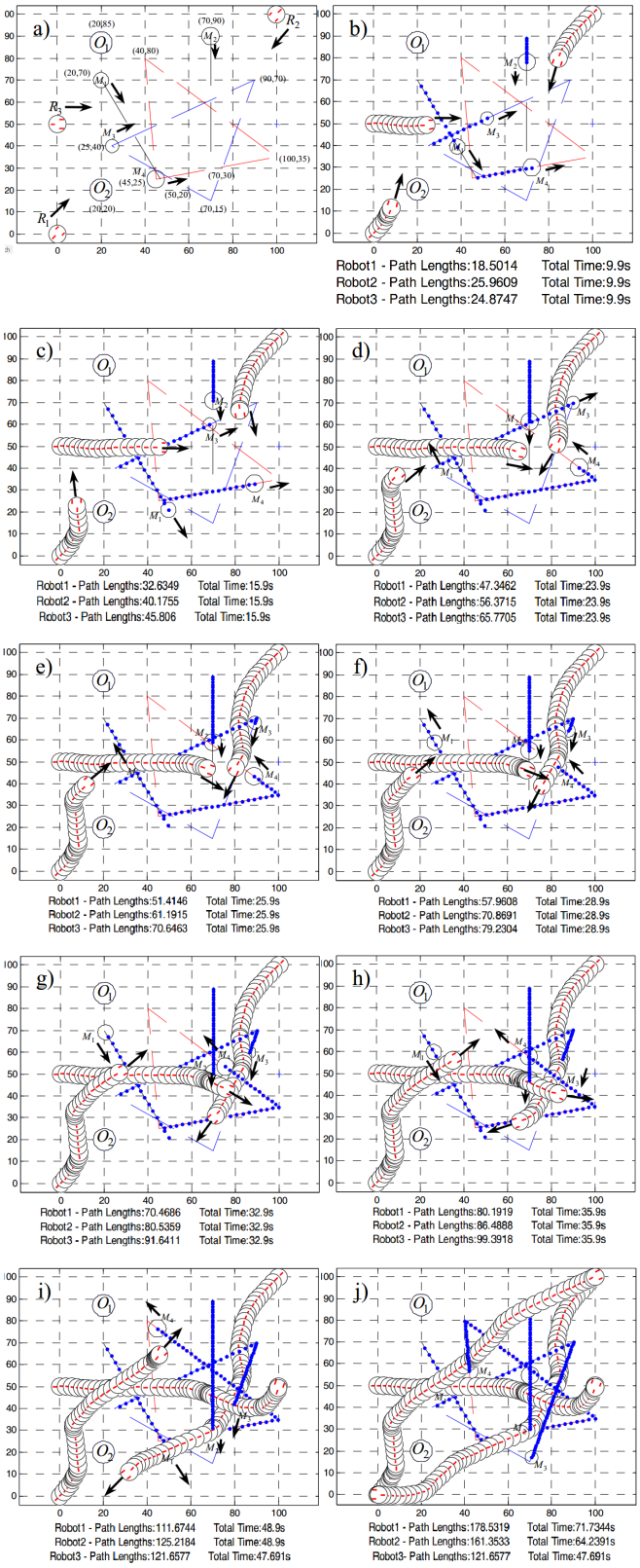
جدول ۳.۳: تعریف موانع پویا

موانع	شعاع‌ها	راس‌های مسیر	سرعت
$M_۱$	۳۰۵	$[۲۰,۷۰] - [۵۰,۲۰]$	۰.۹
$M_۲$	۴	$[۷۰,۹۰] - [۷۰,۳۰]$	۰.۳
$M_۳$	۳	$[۲۵,۴۰] - [۹۰,۷۰], [۷۰,۱۵]$	۰.۷۵ ⟶ ۰.۳ ⟶ ۰.۳۵
$M_۴$	۴	$[۴۵,۲۵] - [۱۰۰,۳۵] - [۴۰,۸۰]$	۰.۷ ⟶ ۰.۶ ⟶ ۰.۳

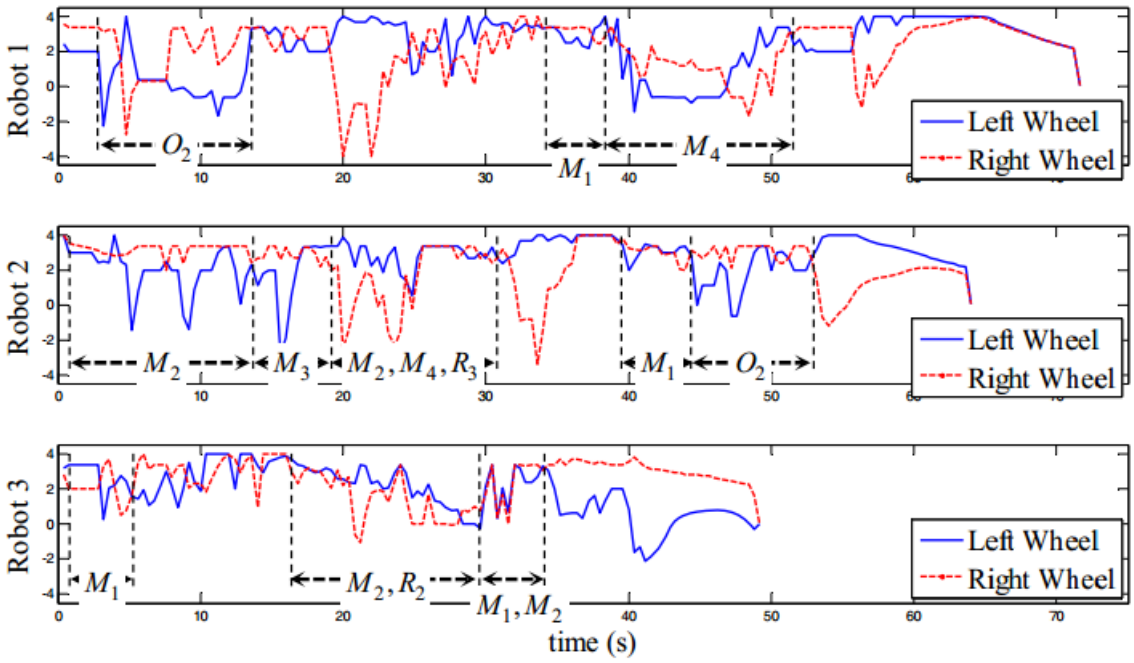
۳.۵.۳ نتایج در محیط موانع پویا

نتایج شبیه‌سازی در محیط پویا در شکل ۱۲.۳ نشان داده‌شده است. در اینجا، دو مانع ایستا و چهار مانع پویا وجود دارد. موانع پویا شعاع مختلف، مسیرهای حرکت و سرعت متغیرها را ارائه می‌دهند و در طی فرآیند حرکت به‌خوبی بین نقاط خاص حرکت می‌کنند. همان‌طور که در شکل ۱۲.۳ (a) نشان داده‌شده است، مسیرهای تمام موانع پویا به‌صورت خط تیره نشان داده‌شده است. در این شبیه‌سازی، موانع پویا $M_۱$ و $M_۲$ درامتداد مسیرهای مستقیم به حرکت درمی‌آیند و $M_۳$ و $M_۴$در طول مسیرهای مثلثی تشکیل‌شده توسط سه رأس به ترتیب با سه سرعت مختلف بین سه رأس حرکت می‌کنند. مختصات دامنه حرکتی برای موانع پویا و سرعت مربوطه به‌صورت جدول ۳.۳ تعریف شده است. در اینجا سرعت به‌عنوان حداکثر سرعت خطی ربات تعریف می‌شود.

همانطور که در شکل ۱۲.۳ نشان داده‌شده است، ربات‌ها حرکت را از کادر a شروع می‌کنند و در کادر j به پایان می‌رسانند. شکل‌های دیگر لحظات اجتناب از موانع را توصیف می‌کنند و پیکان سیاه نشان‌دهنده جهت حرکت ربات‌ها و موانع در لحظات مربوطه است. به‌ویژه اگر ربات برای جلوگیری از برخورد فقط با چرخش به چپ یا راست، غیرممکن باشد، تازمان توقف کامل، سرعتش کم خواهدشد (به‌عنوان‌مثال $R_۱$ در کادر i و $R_۳$ در کادر f و غیره). در بعضی موارد، همچنین می‌تواند قادر به حرکت به‌عقب باشد. در شرایط پیچیده، ربات تنها از منطق فازی برای اجتناب از موانع استفاده نمی‌کند بلکه نیاز به کاربرد قوانین ذکرشده در ۴.۳ دارد. به‌عنوان مثال، همان‌طور که در شکل ۱۲.۳ d تا h نشان داده‌شده است، سه مانع و دو ربات $R_۲$ و $R_۳$ به سرعت در حال نزدیک شدن به یکدیگرند. ابتدا، درکادر d، هر دو ربات برای کاهش حرکت برخورد، سرعت خود را سریعا کاهش می‌دهند، سپس در کادر e، $R_۳$، به حرکت مستقیم با سرعت کم ادامه می‌دهد و $R_۲$ با سرعت هدایت می‌شود. پس ازآن‌که $R_۲$ از میدان خطر f عبور کرد، $R_۳$ به سرعت از این منطقه عبور می‌کند. علاوه بر این، منحنی‌های تغییر سرعت برای تمام ربات‌ها همان‌طور که در شکل ۱۳.۳ توصیف شده‌اند و مناطق متناظر با اجتناب از مانع نیز در این شکل مشخص‌شده‌اند.



شكل ۱۲.۳: نتايج در محيط موانع پويا



شکل ۱۳.۳: منحنی تغییر سرعت چرخ‌های چپ و راست در محیط با موانع پویا

فصل ۴

الگوریتم پیشنهادی مسیریابی برای ربات‌ها در محیط پویای ناشناخته خلوت

۱.۴ مقدمه

در این فصل، الگوریتم پیشنهادی خود برای مسیریابی مبتنی بر منطق فازی برای ربات‌های متحرک در محیط‌های پویای ناشناخته را شرح می‌دهیم. تاکنون الگوریتم‌های مختلفی برای این منظور توسط پژوهشگران توسعه داده شده است که در آنها معمولاً ربات از اطلاعات محلی برای مسیریابی 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. بنابراین حسگرها این اطلاعات را در اختیار پردازنده قرار می‌دهند و ربات در هر گام باید یکی از حفره‌های بین موانع را برای عبور انتخاب کند. اولویت‌بندی و انتخاب حفره برای عبور، کاری است که کنترل‌کننده فازی آن را انجام می‌دهد. پس از انتخاب حفره، فرمان چرخش و حرکت به عملگرهای ربات داده می‌شود و تا نقطه موردنظر جابجا می‌شود. این کارها تا زمانی که ربات به مقصد برسد، تکرار می‌شود. در این روش عمل مسیریابی (انتخاب حفره برای عبور برای رسیدن به مقصد و همچنین اجتناب از موانع) را یک کنترل‌کننده فازی انجام می‌دهد و

ما در این فصل کنترل‌کننده فازی پیشنهادی خود را شرح می‌دهیم و سپس نتایج پیاده‌سازی و اجرای آن در نرم‌افزار متلب را مورد بحث و بررسی قرار می‌دهیم.

۲.۴ بیان مسئله

محیط مورد جستجو 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. هدف نخست الگوریتم پیشنهادی، کمینه کردن مسیر طی شده ربات‌ها و هدف دوم، کمینه کردن زمان رسیدن به مقصد است. فرض بر این است که تعداد ربات‌ها در مقایسه با مساحت محدوده، خلوت است. محدوده : یک مستطیل به طول Le و عرض We است که ربات‌ها و موانع داخل آن تعریف می‌شوند و اطراف آن حصار غیرقابل عبور وجود دارد. ربات : در اینجا بدون از دست دادن کلیت مسئله، ربات‌ها را بصورت مربعی با ضلع یک متر در نظر می‌گیریم. همچنین هر ربات یک زاویه دید ۱۸۰ 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 ممکن از مبدأ به مقصد موجود است و باید پیمایش مسیر، بدون برخورد باشد. اطلاعات ورودی هر ربات محدود به مختصات مکان فعلی خود ربات، مختصات مقصد و مختصات موانع قابل رؤیت است که از طریق حسگرها تخمین زده می‌شود. مجموعه کل ربات‌ها را با R نشان می‌دهیم. مانع ایستا : یک چند ضلعی غیرشفاف و غیرقابل نفوذ که مختصات رئوس آن در طول اجرای الگوریتم غیرقابل تغییر است. مانع پویا : یک چند ضلعی غیرشفاف و غیرقابل نفوذ که مختصات رئوس آن در طول اجرای الگوریتم قابل تغییر است. در اینجا ربات‌ها نسبت به یکدیگر مانع پویا محسوب می‌شوند.

۳.۴ الگوریتم پیشنهادی

همه ربات‌ها از روش یکسان برای مسیریابی استفاده می‌کنند و ما در این بخش الگوریتم یک ربات را شرح می‌دهیم.

۱.۳.۴ استراتژی کلی حرکت

الگوریتم ۱.۴ استراتژی هر ربات برای حرکت از مبدأ به مقصد را نمایش می‌دهد. این الگوریتم شامل دو بخش ’’پویش و تصمیم‌گیری’’ و ’’حرکت’’ است. هنگامی که ربات در مسیر خود

یک مانع متحرک مشاهده می‌کند، به مدت 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 درنظر گرفته‌ایم، پس هیچگاه یک ربات مدت زمان زیادی را منتظر نمی‌ماند. زمانی که مانع متحرکی در مسیر ربات نباشد، حفره‌های پیش رو را اولویت‌بندی کرده و سپس با سرعت ثابت به سمت حفره‌ای که بیشترین اولویت را دارد (مقصد موقت فعلی) حرکت می‌کند. سرعت ربات‌ها ممکن است با هم متفاوت باشد. 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.

۲.۳.۴ کنترل‌کننده فازی

در الگوریتم ۱.۴ ، دستور (۱ـ۵ـ۲) مربوط به محاسبه اولویت‌حفره انتخابی برای عبور است که با استفاده از کنترل‌کننده فازی انجام می‌شود. در ادامه این بخش، ورودی‌ها، خروجی و ساختار این کنترل‌کننده فازی را شرح می‌دهیم. شکل ۲.۴ اطلاعات موردنیاز برای محاسبه اولویت یک حفره را نشان می‌دهد. زاویه θ 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 θ درجه انحراف خواهد داشت. بنابراین هرچه این زاویه بزرگتر باشد، مسیر انحراف بیشتری دارد 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. کنترل‌کننده فازی از دو پارامتر زاویه θ (در پیاده‌سازی آن را angle می‌نامیم) و مسافت D (در پیاده‌سازی آن را distance می‌نامیم) برای محاسبه اولویت حفره موردنظر استفاده می‌کند. در ادامه این بخش، متغیرهای زبانی متناظر با این دو پارامتر، توابع عضویت آنها و همچنین قواعد ''اگر‌ـ آنگاه'' فازی برای استنتاج میزان اولویت p و همچنین روش غیرفازی سازی مورد استفاده را شرح می‌دهیم. شکل ۳.۴ شمای کلی کنترل‌کننده فازی طراحی شده را نمایش می‌دهد که مقدار خروجی (Priority) با استفاده از ۲۵ قاعده استنتاج و با منطق فازی ممدانی از ورودی‌ها (angle و distance) به دست می‌آید. مقدار زاویه θ در بازه $[0^\circ, 180^\circ]$ قرار دارد زیرا اگر خط واصل ربات به مقصد آن را بعنوان محور در نظر بگیریم، در حالت کلی انحراف به سمت چپ

سرعت ربات r : v_r مختصات فعلی ربات r : $[x_r,y_r]$

مختصات مقصد d : $[x_d,y_d]$ مختصات حفره h : $[x_h,y_h]$

اولویت حفره h : P_h

۱- پویش و تصمیم‌گیری :

۱-۱- اگر $[x_r,y_r]=[x_r,y_r]$ آنگاه برو به مرحله ۳

۱-۲- تا زمانی که مانع متحرکی در فاصله نزدیک است:

۱-۲-۱- یک عدد تصادفی $t_r \in [1,3]$ انتخاب کن

۱-۲-۲- به مدت t_r ثانیه صبر کن

۱-۳- مقدار اولویت را قرار بده $P \leftarrow 0$

۱-۴- مقدار مکان بعدی را قرار بده $[x'_r,y'_r] \leftarrow [x_r,y_r]$

۱-۵- برای هر حفره قابل رؤیت h :

۱-۵-۱- محاسبه مختصات $[x_h,y_h]$

۱-۵-۲- محاسبه مقدار اولویت P_h (کنترل‌کننده فازی)

۱-۵-۳- اگر $P_h > P$: $P \leftarrow P_h$ و $[x'_r,y'_r] \leftarrow [x_h,y_h]$

۲- حرکت:

۲-۱- با سرعت v_r به سمت $[x'_r,y'_r]$ حرکت کن

۲-۲- اگر به مانع متحرک نزدیک شدی برو به مرحله ۱

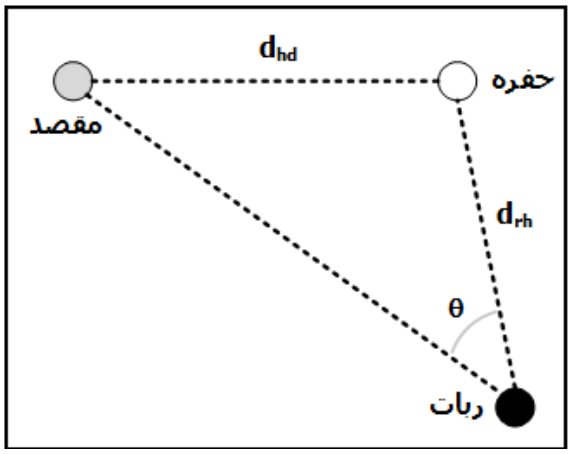
۲-۳- اگر به $[x'_r,y'_r]$ رسیدی برو به مرحله ۱

۳- پایان.

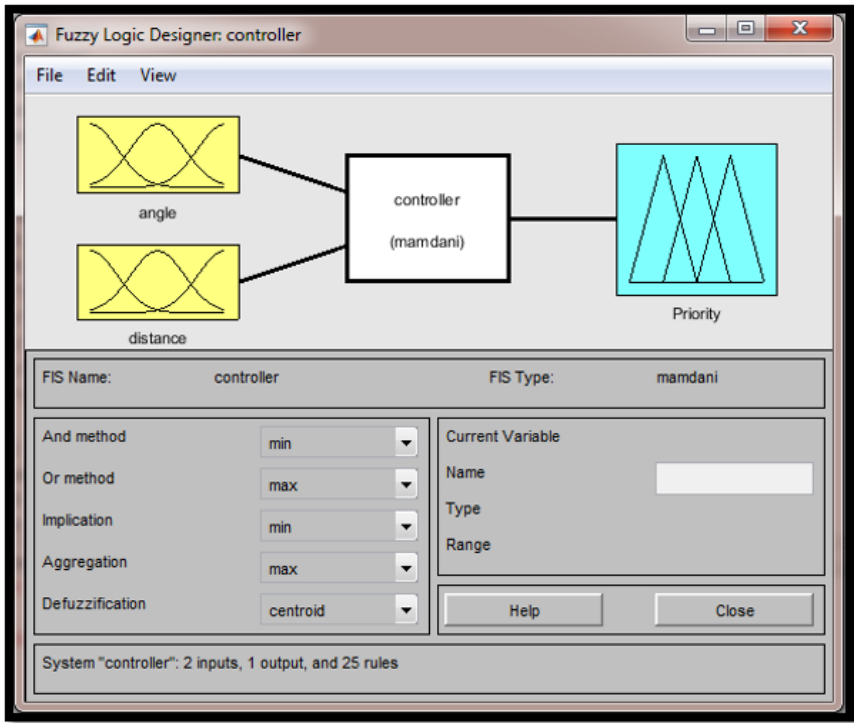
شکل ۱.۴: استراتژی حرکت برای یک ربات $r \in R$

یا راست آن فرقی ندارد. متغیرهای زبانی مربوط به زاویه θ عبارتند از : خیلی تند (VS) ، تند (S)، متوسط (M) ، باز (B) ، و خیلی باز (VB) . شکل ۴.۴ توابع عضویت مربوط به این مقادیر را نمایش می‌دهد.

مقدار مسافت D در بازه $[\circ, Le + We]$ 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

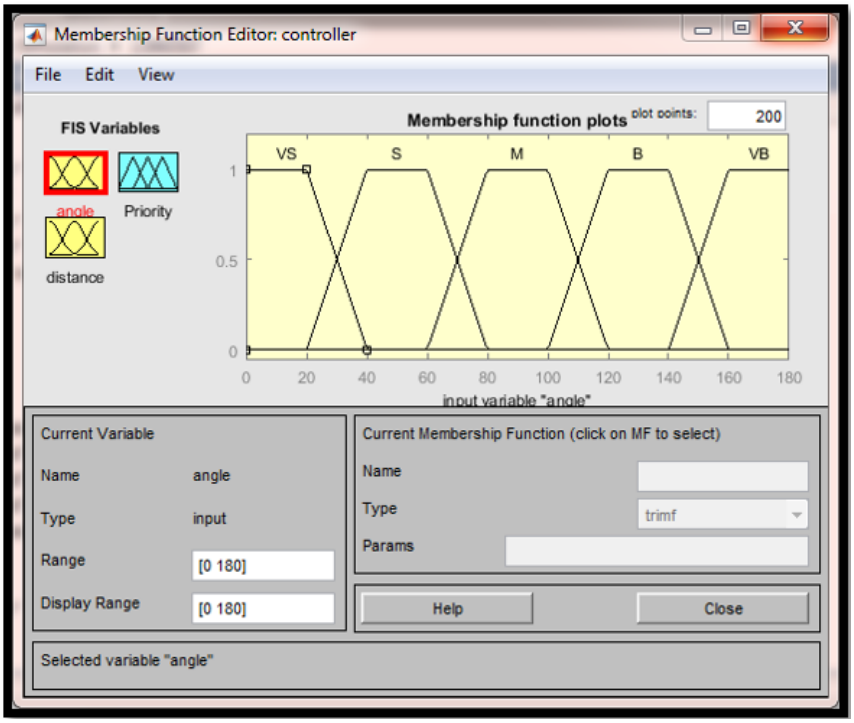


شکل ۲.۴: مثالی از وضعیت نسبی یک ربات و مقصد آن با یک حفره

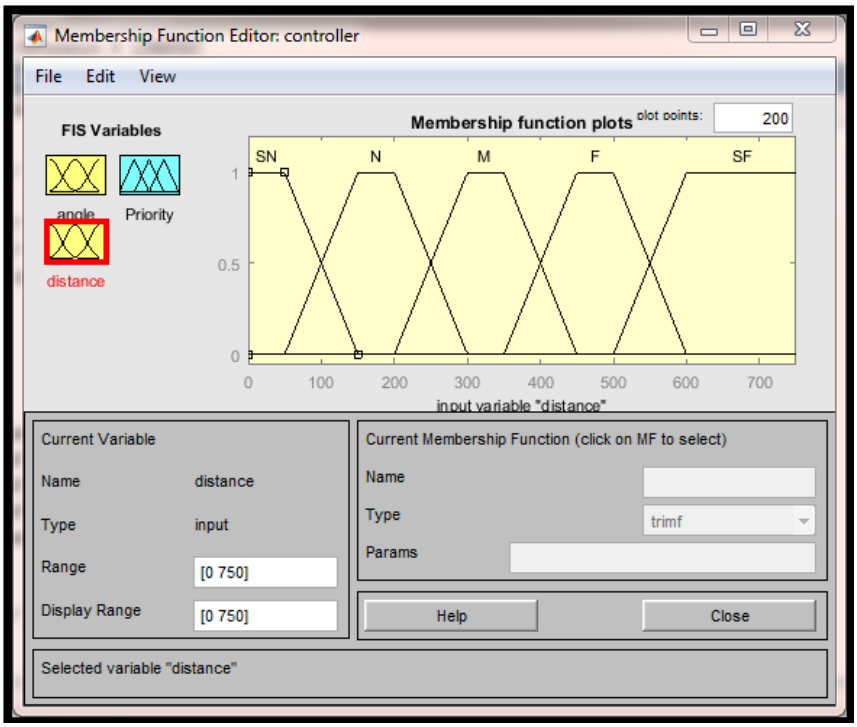


شکل ۳.۴: شمای کلی کنترل‌کننده فازی

محدوده باشد. در این صورت ربات برای عبور از حفره و سپس رسیدن به مقصد، در بهترین حالت باید مسافتی به اندازه مجموع طول و عرض محدوده را طی کند. متغیرهای زبانی مربوط به مسافت D عبارتند از : خیلی نزدیک (VS) ، نزدیک (S) ، متوسط (M) ، دور (B) و خیلی دور (VB) . شکل ۵.۴ توابع عضویت مربوط به این مقادیر زبانی را برای بازه حقیقی $[۰,۷۵۰]$ نمایش می‌دهد. خروجی کنترل فازی، اولویت حفره است. مقدار اولویت P در بازه $[۰,۹]$ قرار دارد به طوری که ۹ بالاترین و ۰ پایین‌ترین اولویت است. پس از آن‌که ربات در مرحله پویش

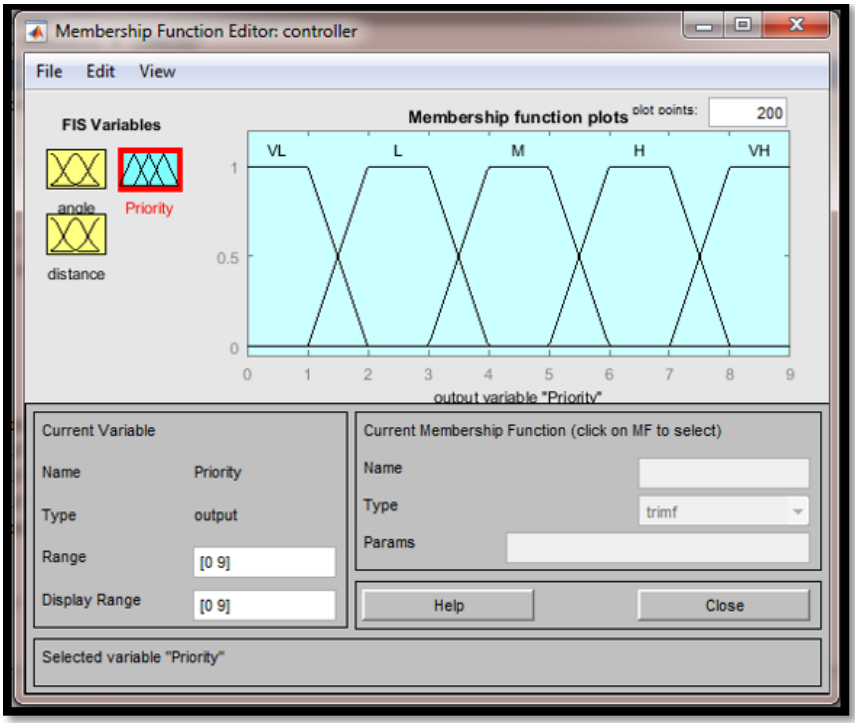


شکل ۴.۴: توابع عضویت مربوط به متغیرهای زبانی زاویه θ



شکل ۵.۴: توابع عضویت مربوط به متغیرهای زبانی مسافت D

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 متغیرهای زبانی متناظر با اولویت پشتیبانی می‌کنند و پس از استنتاج، با استفاده از عمل غیرفازی سازی، یکی از مقادیر صحیح ۰ تا ۹ بعنوان خروجی بدست می‌آید. متغیرهای زبانی مربوط به اولویت p عبارتند از : خیلی بالا (VH) ، بالا (H) ، متوسط (M) ، پایین (L) و خیلی پایین (VL) . شکل ۶.۴ توابع عضویت مربوط به این مقادیر زبانی را نمایش می‌دهد.



شکل ۶.۴: توابع عضویت مربوط به متغیرهای زبانی اولویت P

از آنجا که دو متغیر ورودی زاویه انحراف و مسافت هر کدام دارای ۵ متغیر زبانی متناظر هستند، بنابراین با ۲۵ قاعده استنتاج می‌توان همه حالت‌های مختلف ورودی را پوشش داد. قواعد استنتاج تعریف شده در جدول ۱.۴ با هدف هدایت ربات در مسیر تقریباً مستقیم و جلوگیری از انحراف زیاد طراحی شده است. به عنوان مثال، ستون VS شامل سه اولویت ”خیلی بالا“ و دو اولویت ”بالا“ است اما در ستون‌های بعدی که مربوط بزرگتر شدن زاویه انحراف است، میزان اولویت‌ها کاهش می‌یابد. همین وضعیت برای سطرها اتفاق افتاده است؛ یعنی در هر ستون، هرچه شماره سطر کمتر باشد (مسافت کمتر باشد) اولویت بیشتر است و با افزایش مسافت تخمینی برای عبور از حفره مورد نظر، اولویت آن حفره کاهش می‌یابد.

جدول ۱.۴: قوانین استنتاج اولویت در کنترل‌کننده فازی پیشنهادی

	P	θ				
		VS	**S**	**M**	**B**	**VB**
D	**SN**	VH	VH	H	M	L
	N	VH	H	M	L	L
	M	VH	H	M	L	VL
	F	H	H	L	VL	VL
	SF	H	M	L	VL	VL

۴.۴ نتایج شبیه‌سازی

در این بخش نتایج شبیه‌سازی الگوریتم پیشنهادی را مورد بررسی و تحلیل قرار می‌دهیم و آن را با الگوریتم ۷.۳ مقایسه خواهیم کرد.

۱.۴.۴ محیط پیاده‌سازی و مقداردهی پارامترها

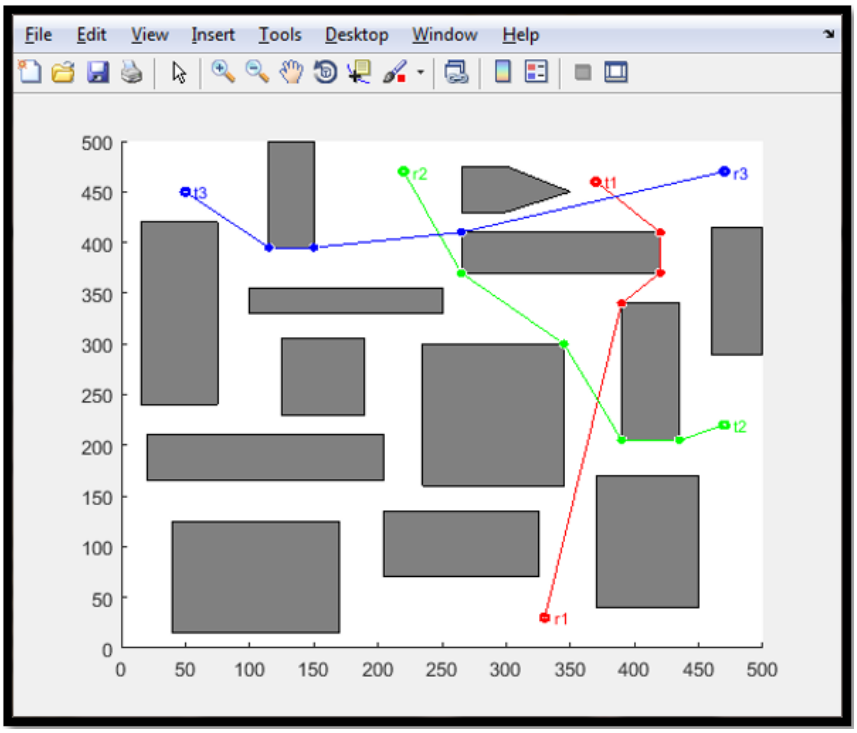
الگوریتم ۱.۴ در محیط نرم‌افزار متلب پیاده سازی و کنترل‌کننده فازی آن نیز توسط جعبه ابزار متلب آماده شده است. پارامترهای مسئله مانند محدوده، مبدأ، مقصد و سرعت ربات‌ها و همچنین مختصات موانع توسط یک فایل متنی به‌عنوان ورودی به برنامه داده می‌شود. پس از اجرای کد برنامه، ابتدا مراحل پیش پردازش مانند شبیه‌سازی محیط انجام شده و سپس الگوریتم ۱.۴ اجرا می‌شود. خروجی برنامه، مکان‌های توقف، مدت توقف و مسیری است که هر ربات باید طی کند تا به مقصد برسد. برای مرحله پردازش و تصمیم‌گیری زمان صفر در نظر گرفته شده است؛ بنابراین تعداد گام‌هایی که ربات از مبدأ تا مقصد طی می‌کند روی زمان حرکت آن تأثیر ندارد و در اینجا منظور از زمان‌های توقف، زمان‌هایی است که ربات به خاطر دیدن یک مانع متحرک در جای خود ثابت می‌ماند. 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 ایستا لزوماً خلوت نیست. برای هر اجرای الگوریتم، تعداد مشخصی ربات با مبدأ و مقصد تصادفی و همچنین سرعت تصادفی بین ۱ تا ۵ واحد در ثانیه تولید شده و این مشخصات محیط برای اجرای الگوریتم پیشنهادی و همچنین الگوریتم ۷.۳ یکسان در نظر گرفته شده است. تعداد ربات‌ها در هر آزمایش از ۱ تا ۱۰ دستگاه درنظر 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.

۲.۴.۴ تصمیم‌گیری در مورد استراتژی و پارامترهای ورودی کنترل‌کننده فازی

در الگوریتم ۱.۴ هنگامی که ربات با مانع پویا روبرو می‌شود استراتژی ایست موقت را اجرا می‌کند اما الگوریتم ۷.۳ تغییر مسیر ربات را بعنوان استراتژی اجتناب از موانع پویا در نظر گرفته است. همچنین موارد اساسی دیگر، پارامتر مسافت D ، مجموعه متغیرهای زبانی، توابع عضویت و مجموعه قواعد استنتاج است. پارامتر زاویه انحراف در هر دو الگوریتم یکسان است. در الگوریتم ۱.۴ پارامتر مسافت D مجموع فاصله ربات تا حفره و فاصله حفره تا مقصد است اما در الگوریتم ۷.۳ مقدار این پارامتر برابر است با فاصله ربات تا مقصد. بنابراین ما سه نسخه دیگر از الگوریتم ۱.۴ ارائه دادیم و استراتژی برخورد با مانع پویا یا پارامتر مسافت D را با الگوریتم ۷.۳ یکسان در نظر گرفتیم و آزمایش‌های مختلفی را انجام دادیم تا از سازگاری استراتژی پیشنهادی، پارامتر مسافت D و مجموعه قواعد استنتاج اطمینان حاصل کنیم. سه نسخه آزمایشی جدید از الگوریتم ۱.۴ عبارتند از: رویکرد II : در این نسخه، تنها تفاوت با الگوریتم اصلی، پارامتر مسافت D است که مانند الگوریتم ۷.۳ فاصله مستقیم ربات تا مقصد در نظر گرفته شده است. رویکرد III : تفاوت این رویکرد با نسخه اصلی، استراتژی اجتناب از مانع پویا است که مانند الگوریتم ۷.۳ از استراتژی ''تغییر مسیر'' استفاده شده است. رویکرد IV : در این رویکرد، هم پارامتر مسافت D و هم استراتژی اجتناب از مانع پویا مانند الگوریتم ۷.۳ در نظر گرفته شده است. اکنون نتایج به‌دست آمده از آزمایش‌های مختلف روی نسخه اصلی الگوریتم ۱.۴ و سه رویکرد دیگر را در قالب مثالی از یک محیط آزمایشی مورد بحث و بررسی قرار می‌دهیم. در شکل ۷.۴ مسیر پیشنهادی الگوریتم ۱.۴ برای یک محیط آزمایشی مشاهده می‌شود. ربات‌ها با r_i و مقصد آنها با t_i مشخص شده است. ربات $r_۲$ در گام سوم برای اجتناب از برخورد با ربات $r_۱$ به مدت ۲ ثانیه توقف داشته و در طول مسیر اتفاق مشابهی برای ربات‌ها رخ نداده است و هرکدام مسیر مناسبی را طی کرده‌اند. همچنین در آزمایش‌های مختلف، مسیر پیشنهادی الگوریتم ۱.۴ همواره بهینه یا نزدیک به بهینه بوده است.

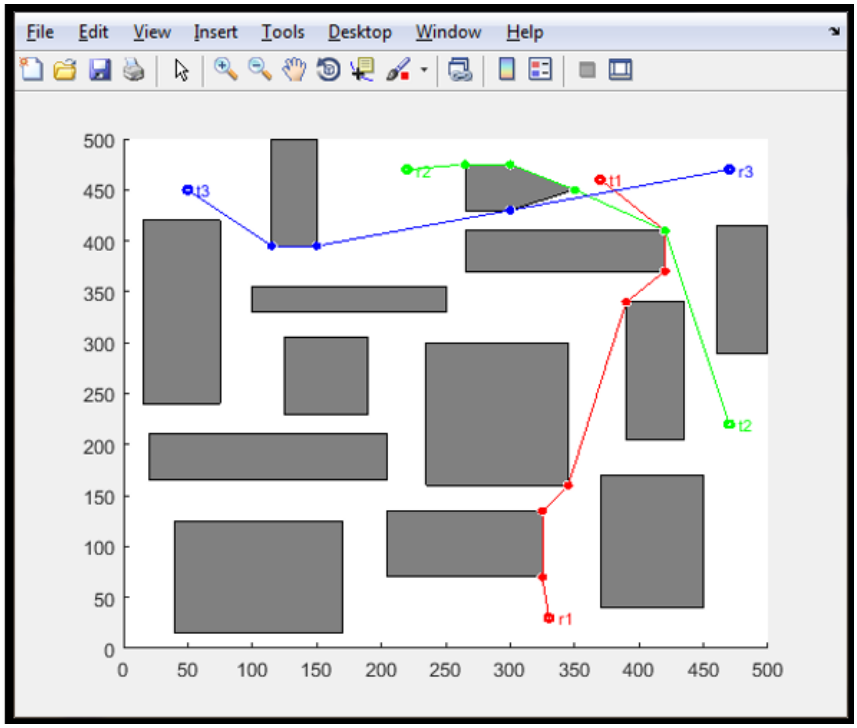
در شکل ۸.۴ مسیر پیشنهادی رویکرد II برای یک محیط آزمایشی مشاهده می‌شود. در اینجا مسیر ربات‌های $r_۱$ و $r_۳$ نسبت به مسیر پیشنهادی الگوریتم ۱.۴ تغییر اندکی داشته و مسیر $r_۲$ کاملاً متفاوت است. همچنین ربات $r_۳$ در گام نخست برای اجتناب از برخورد با ربات $r_۲$ به مدت ۳ ثانیه و ربات $r_۱$ نیز در گام ششم برای اجتناب از برخورد با ربات $r_۲$ به مدت ۲ ثانیه توقف داشته‌اند. نکته قابل توجه در اینجا، رفتار ربات $r_۱$ در گام‌های اول تا سوم است. این ربات در ابتدای مسیر فقط با موانع ایستا مواجه شده و در تصمیم‌گیری برای مسیر بهینه اندکی دچار اختلال شده است. رویکرد II نسبت به الگوریتم ۱.۴ در این محیط



شکل ۷.۴: مسیر پیشنهادی الگوریتم ۱.۴ برای سه ربات با مبدأ، مقصد و سرعت تصادفی در یک محیط آزمایشی

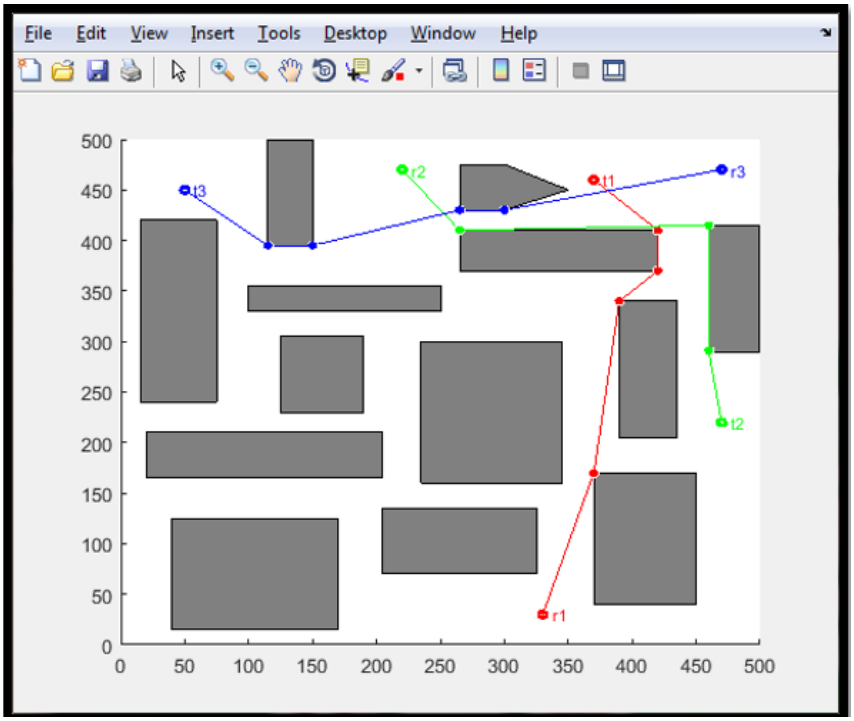
آزمایشی به ترتیب حدود ۷٪ و ۵٪ پنالتی زمان و طول مسیر طی شده داشته است. در شکل ۹.۴ مسیر پیشنهادی رویکرد III برای یک محیط آزمایشی مشاهده می‌شود. در اینجا مسیر ربات r_1 نسبت به مسیر پیشنهادی الگوریتم ۱.۴ تغییر اندکی داشته و مسیر r_2 کاملاً متفاوت است. در این رویکرد، ربات‌ها برای اجتناب از برخورد با یکدیگر توقف ندارند بلکه تغییر مسیر می‌دهند. ربات r_1 در مسیر خود از طرف سایر ربات‌ها دچار مزاحمت نمی‌شود و مسیر آن با مسیر پیشنهادی الگوریتم ۱.۴ یکسان است. ربات r_3 در گام سوم ربات r_2 را مشاهده می‌کند و با اینکه ربات r_2 در حال دور شدن از ربات r_3 است، باعث تغییر مسیر ربات r_3 و افزایش مسافت کل شده است. در این آزمایش، سرعت ربات r_2 از سرعت ربات r_1 کمتر بوده و در اواسط مسیر خود ربات r_1 را مشاهده می‌کند و مسیر خود را تغییر می‌دهد که باعث افزایش چشمگیر مسافت کل شده است. رویکرد III نسبت به الگوریتم ۱.۴ در این محیط آزمایشی به ترتیب حدود ۶٪ و ۷٪ پنالتی زمان و طول مسیر طی شده داشته است.

در شکل ۱۰.۴ مسیر پیشنهادی رویکرد IV برای یک محیط آزمایشی مشاهده می‌شود. در این رویکرد، ربات‌ها برای اجتناب از برخورد با یکدیگر توقف ندارند بلکه تغییر مسیر می‌دهند. همچنین پارامتر مسافت D نیز فاصله ربات تا مقصد در نظر گرفته شده است. در اینجا مسیر ربات r_1 با مسیر پیشنهادی الگوریتم ۱.۴ یکسان است اما مسیر دو ربات دیگر با مقداری خرابی مواجه شده است. مسیر حرکت ربات r_1 در الگوریتم ۱.۴ و رویکرد IV تقریباً مشابه است و به



شکل ۸.۴: مسیر پیشنهادی رویکرد II برای سه ربات با مبدأ، مقصد و سرعت تصادفی در یک محیط آزمایشی

نظر می‌رسد که خرابی نسبی آن در نتیجه تأثیر پارامتر مسافت D باشد. ربات $r_۲$ در رویکرد IV مسیری تقریباً مشابه الگوریتم ۱.۴ را پیموده اما در گام‌های دوم و چهارم دچار اولویت‌بندی آشکار در انتخاب حفره بعدی شده است. با توجه به رفتار این ربات در رویکردهای قبلی، به نظر می‌رسد که ترکیب استراتژی ”تغییر مسیر برای اجتناب از موانع پویا“ با تأثیر ”پارامتر مسافت D“ منجر به این انتخاب غیربهینه شده است؛ هرچند بصورت شهودی نمی‌توان در شکل نهایی مسیر حرکت ربات‌ها چنین چیزی را مستقیماً استنتاج کرد. حتی گام پنجم ربات $r_۲$ نیز منطقی به نظر نمی‌رسد. شاید اینگونه به نظر برسد که با اضافه کردن قید ”در صورت مشاهده مقصد، مستقیماً به سمت آن حرکت کن“ به این رویکرد، مشکل برطرف می‌شود اما باید این نکته را در نظر بگیریم که ربات در اینجا یک مانع پویا را مشاهده کرده و در نتیجه آن تصمیم نادرست گرفته است. بنابراین باید استراتژی چرخش به درستی تنظیم شود یا اینکه با استفاده از حسگرهای بیشتر (مانند الگوریتم ۷.۳) سرعت و جهت حرکت مانع پویا را اندازه‌گیری نموده و امکان برخورد را به صورت دقیق‌تر به دست آوریم. رویکرد IV نسبت به الگوریتم ۱.۴ در این محیط آزمایشی به ترتیب حدود ۶٪ و ۹٪ پنالتی زمان و طول مسیر طی شده داشته است. علاوه بر اجرای آزمایش‌های مختلف در این محیط آزمایشی، تعدادی محیط تصادفی دیگر نیز تولید و آزمایش‌های مشابه در آنها تکرار شد و برای انتخاب استراتژی اجتناب از مانع پویا و همچنین پارامتر مسافت D آمارها در الگوریتم ۱.۴ از سایر رویکردهای



شکل ۹.۴: مسیر پیشنهادی رویکرد III برای سه ربات با مبدأ، مقصد و سرعت تصادفی در یک محیط آزمایشی

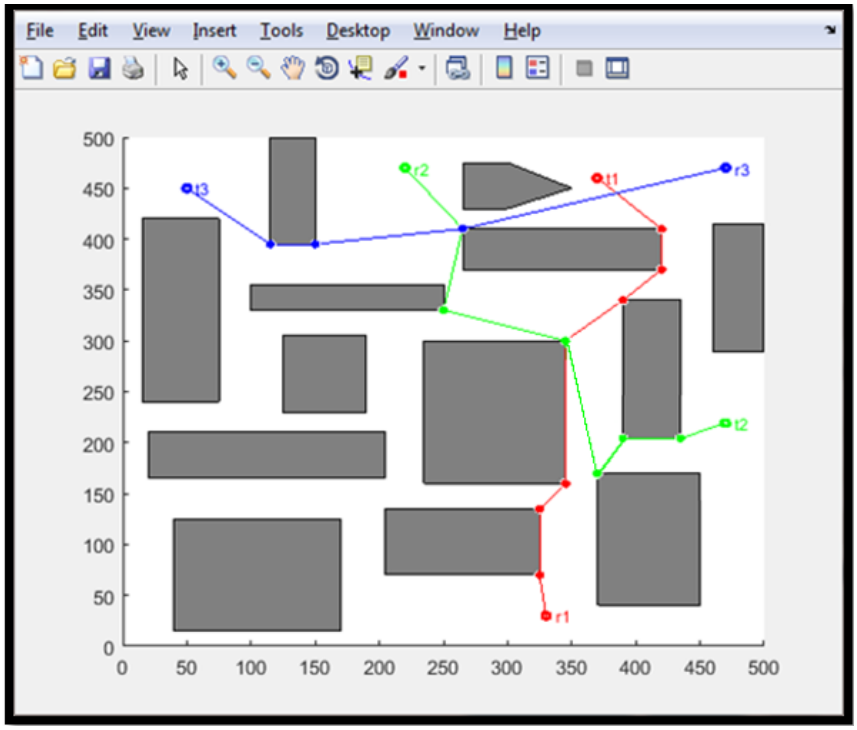
آن بهتر بود.

۳.۴.۴ نتایج تجربی پیاده‌سازی الگوریتم پیشنهادی

اهداف بهینه‌سازی در الگوریتم پیشنهادی، ”کمینه سازی زمان رسیدن ربات‌ها به مقصد” و ”کمینه سازی مسافت طی شده توسط ربات‌ها” هستند. در این بخش نتایج تجربی پیاده سازی الگوریتم را برای این دو هدف مورد بررسی قرار می‌دهیم.

جدول ۲.۴ میانگین زمان رسیدن ربات‌ها به مقصد را نمایش می‌دهد. همانطور که مشاهده می‌شود، الگوریتم پیشنهادی برای هدف ”کمینه سازی زمان رسیدن ربات به مقصد” کارآیی بالاتری نسبت به الگوریتم ۷.۳ دارد. همچنین زمانی که فقط یک ربات در محیط وجود دارد (ربات پویا در محیط نباشد)، الگوریتم پیشنهادی بهتر عمل کرده و نشان می‌دهد که برای محیط‌های ایستا نیز مناسب‌تر است.

جدول ۳.۴ میانگین مسافت طی شده توسط ربات‌ها برای رسیدن به مقصد را نمایش می‌دهد. همانطور که مشاهده می‌شود، برای هدف ”کمینه سازی مسافت طی شده تا مقصد” نیز الگوریتم پیشنهادی کارآیی بالاتری نسبت به الگوریتم ۷.۳ دارد. بنابراین یک توقف کوتاه نتایج بهتری از تغییر مسیر آنی دارد. در اینجا نیز مشاهده می‌شود که در مورد مسافت طی شده تا مقصد برای زمانی است که فقط یک ربات در محیط وجود دارد، الگوریتم پیشنهادی



شکل ۴.۱۰: مسیر پیشنهادی رویکرد IV برای سه ربات با مبدأ، مقصد و سرعت تصادفی در یک محیط آزمایشی

بهتر عمل کرده و برای محیط‌های ایستا مناسب‌تر است.

۵.۴ نتیجه‌گیری

در این فصل یک الگوریتم مسیریابی برای ربات‌های متحرک در محیط‌های پویای ناشناخته خلوت ارائه شده است که با استفاده از منطق فازی، ربات را در هرگام به سمت مقصد هدایت می‌کند. در این الگوریتم یک کنترل‌کننده فازی عمل اولویت‌بندی حفره‌های مابین موانع (برای عبور) را انجام می‌دهد. نتایج شبیه‌سازی، کارآیی الگوریتم پیشنهادی برای کمینه سازی 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.

جدول ۲.۴: مقایسه زمان رسیدن ربات‌ها به مقصد

تعداد ربات‌ها	میانگین زمان رسیدن به مقصد (ثانیه)	
	الگوریتم ۷.۳	الگوریتم پیشنهادی
۱	۷۴	۵۳
۲	۷۷	۵۵
۳	۷۹	۵۶
۴	۸۶	۵۸
۵	۹۵	۶۲
۶	۱۰۱	۶۵
۷	۱۲۱	۷۰
۸	۱۳۴	۷۱
۹	۱۳۹	۷۴
۱۰	۱۴۷	۷۴

جدول ۳.۴: مقایسه مسافت طی شده توسط ربات‌ها تا مقصد

تعداد ربات‌ها	میانگین مسافت طی شده تا مقصد (ثانیه)	
	الگوریتم ۷.۳	الگوریتم پیشنهادی
۱	۲۲۵	۱۶۲
۲	۲۰۸	۱۴۸
۳	۲۷۶	۱۹۷
۴	۳۱۹	۲۱۴
۵	۳۹۰	۲۵۴
۶	۴۱۴	۲۶۷
۷	۴۳۵	۲۵۲
۸	۴۶۹	۲۵۳
۹	۲۶۲	۲۴۴
۱۰	۴۲۶	۲۱۵

فصل ۵

نتیجه‌گیری

در این پایان‌نامه نخست مسئله برنامه‌ریزی حرکت گروهی از ربات‌ها در محیط‌های ناشناخته پویا را مورد بررسی قرار دادیم و سپس رویکرد استفاده از کنترل‌کننده‌های فازی را تشریح نمودیم. ابتدا نحوه مسیریابی در محیط‌های ناشناخته ایستا و سپس روش مواجه شدن و اجتناب از برخورد با موانع پویا و همچنین حرکت به سوی مقصد با استفاده از استراتژی‌های مختلف حرکت و کنترل‌کننده‌های فازی مورد واکاوی قرار گرفت. کنترل‌کننده‌های فازی در این روش‌ها کار اولویت‌بندی مسیرهای محلی (حفره‌های مابین موانع) برای عبور را انجام می‌دهند و با استفاده از دو ورودی زاویه انحراف و مسیر تخمینی تا مقصد و همچنین با مجموعه‌ای از قواعد استنتاج مبتنی برمنطق فازی، به محاسبه اولویت هر مسیر می‌پردازند. این عمل تا رسیدن ربات به مقصد انجام می‌شود.

یکی از رویکردهای مواجه شدن و اجتناب از برخورد با موانع پویا در محیط، رویکرد تغییر مسیر ربات است که همانگونه که در فصل سوم به آن اشاره شد، ربات می‌تواند از دو کنترل کننده فازی استفاده کند که اولین کنترل‌کننده که دارای اولویت بالاتر است، عملیات اجتناب از مانع را مدیریت می‌کند و در زمان مواجه شدن ربات با موانع متحرک، با استفاده از بردار سرعت خود و آنها میزان خطر را محاسبه می‌کند و به جهت کم خطر منحرف می‌شود. این کنترل‌کننده هیچ دیدی نسبت به مقصد ندارد و وظیفه آن تنها اجتناب از موانع است. زمانی که خطر برخورد ربات را تهدید نکند، کنترل‌کننده دوم، ربات را به سوی مقصد هدایت می‌کند.

در فصل چهارم پایان‌نامه روشی جدید برای حرکت ربات در محیط‌های ناشناخته پویای

خلوت ارائه نمودیم که تنها از یک کنترل‌کننده برای مسیریابی و اجتناب از برخورد استفاده می‌کند و همچنین ربات در هنگام مواجه شدن با موانع متحرک، یک زمان تصادفی کوتاه توقف می‌کند تا مانع متحرک دور شود. نتایج شبیه‌سازی در نرم‌افزار $Matlab$ نشان داد که برای محیط‌های خلوت، رویکرد پیشنهادی کارآمدتر از از رویکرد تغییر جهت است و زمان رسیدن ربات به مقصد و همچنین میزان مسافت طی شده را به میزان چشمگیری کاهش می‌دهد.

مراجع

[1] Ying, Hao (2000)," **Fuzzy control and modeling: analytical foundations and applications**", Wiley-IEEE Press.

[2] Latombe J.C. (1991)," **Robot Motion Planning**", Kluwer Academic Publishers.

[3] Barraquand J., Kavraki L.E., Latombe J.C., Li T.Y., Motwani R., and Raghavan P. (1997)," A random sampling scheme for path planning", International Journal of Robotics Research, 16(6),759–774.

[4] Lozano-Perez T.,(1980), Ph.D. Thesis , "Spatial planning with polyhedral models", MIT Ph.D. Thesis (E.E.)

[5] Athans M. and Falb P.L. (1996)," **Optimal Control, An Introduction to the Theory and Its Applications** ", McGrawHill.

[6] Bertsimas D., John N., and Tsitsiklis N. (1997), " **Introduction to Linear Optimization** ", Athena Scientific.

[7] Hamdy A. T. (1987), "Operations Research, An Introduction", Macmillan Publishing Company, New York, 4th edition

[8] Robertson A., Inalhan G., and Jonathan P. H. (1999) "Spacecraft formation flying control design for the orion mission", Proceedings of AIAA/GNC.

[9] Janglova D. (2004), "Neural Networks in Mobile Robot Motion", International Journal of Advanced Robotic Systems, 15-22 .

[10] Low K. H., Leow W. K. , Ang M. H. Jr. (2002) ,"Integrated Planning and Control of Mobile Robot with Self- organizing Neural Network", in Proceedings of ICRA'02. IEEE International Conference on Robotics and Automation, 3870-3875.

[11] Thomas C. E., Pacheco M. A. C., M. M. and Vellasco B. R. (1999), " **Mobile Robot Path Planning Using Genetic Algorithms in Foundations and Tools for Neural Modeling** ", Springer Berlin/ Heidelberg, 671- 679.

[12] Carelli R., Soria C.M. and Morales B. (2005),"Vision-based Tracking Control for Mobile Robots", in Proceedings of 12th International Conference on Advanced Robotics, 148-152.

[13] Padhy P. K. , Sasaki T., Nakamura S. and Hashimoto H. (2010), "Modeling and Position Control of Mobile Robot", in Proceedings of 11th IEEE international Workshop on Advanced Motion Control, 100-105.

[14] Khanh D. Q., Suh Y. S. (2013), "Mobile Robot Destination Generation by Tracking a Remote Controller Using a Vision-aided Inertial Navigation Algorithm", Journal of Electrical Engineering Technology, 613-620.

[15] Park S. H., Kim G. W. (2014), "Expanded Guide Circle- based Obstacle Avoidance for the Remotely Operated Mobile Robot", Journal of Electrical Engineering Technology, 1034-1042.

[16] Ge S. S., Cui Y. J. (2002), "Dynamic motion planning for mobile robots using potential field method" , Auto- nomous Robots, 207-222.

[17] Van Den Berg J. P., Overmars M. H. (2005),"Roadmap- based motion planning in dynamic environments", Robotics, IEEE Transactions on, 885-897.

[18] Zhang C. G., Xi Y. G. (2003), "Rolling path planning and safety analysis of mobile robot in dynamic uncertain environment", Control Theory Applications, 37-44.

[19] Tiago P., Nascimento A. P., Moreira, A. G. S., Conceição A. Bonarini (2013), "Intelligent state changing applied to multi-robot systems", Robotics and Auto- nomous Systems, 115-124.

[20] Zhong X. Y., Zhong X. G., Peng X. F. (2014), "Velocity- Change-Space-based dynamic motion planning for mobile robots navigation" , Neurocomputing, 153-163.

[21] Parhi D. R., Pradhan S. K., Panda A. K. and Behera R. K. (2009), "The stable and precise motion control for multiple mobile robots," Applied Soft Computing, 477-487.

[22] (1995), "Introduction to Fuzzy Systems", in Proceedings of IEEE Electronic Technology Directions to Year 2000, 94-103.

[23] Abedinia O., Wyns B., Ghasemi A. (2011), ”Robust fuzzy PSS design using ABC”, in: 2011 10th International Conference on Environment and Electrical Engineering (EEEIC), IEEE, pp. 1–4.

[24] Aceves A., Aguilar J. (2006), ”A simplified version of Mamdani’s fuzzy controller: the natural logic controller”, IEEE Trans. Fuzzy Syst, 14 (1), 16–30.

[25] Amador-Angulo L., Castillo O. (2014),” Optimization of the Type-1 and Type-2 fuzzy controller design for the water tank using the Bee Colony Optimization”, in: 2014 IEEE Conference on Norbert Wiener in the 21st Century (21CW), IEEE, pp. 1–8.

[26] Amador-Angulo L., Castillo O. (2014), ” **Comparison of the optimal design of fuzzy controllers for the water tank using ant colony optimization** ”, in: Recent Advances on Hybrid Approaches for Designing Intelligent Systems, Springer International Publishing, pp. 255–273.

[27] Amador-Angulo L., Castillo O., Pulido M. (2013),”Comparison of fuzzy controllers for the water tank with Type-1 and Type-2 fuzzy logic”, in: IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS), 2013 Joint, IEEE, pp. 1062–1067.

[28] Baykasoglu A., Özbakyr L., Tapkan P. (2007), ”Artificial bee colony algorithm and its application to generalized assignment problem”, in: T.S.C. Felix, K.T. Manoj (Eds.), Swarm Intelligence: Focus on Ant and Particle Swarm Optimization, Itech Education and Publishing, Vienna, Austria, pp. 113–143.

[29] Bhushan B., Valluru S.K., Singh M. (2013), ”Takagi-Sugeno fuzzy system based stable direct adaptive control of nonlinear systems”, Int J. Comput. Appl, 68 (15) , 30–36.

[30] Caraveo C. , Castillo O. (2014), ” **Optimization of fuzzy controllers design using the bee colony algorithm** ”, in: Recent Advances on Hybrid Approaches for Designing Intelligent Systems, Springer International Publishing, pp. 163–175.

[31] Castillo O., Martínez-Marroquín R., Melin P. , Valdez F., Soria J. (2012),” Comparative study of bio-inspired algorithms applied to the optimization of type-1 and type- 2 fuzzy controllers for an autonomous mobile robot ”, Inf. Sci, 192, 19–38.

[32] Castillo O., Neyoy H., Soria J., García M., Valdez F. (2013),” Dynamic fuzzy logic parameter tuning for ACO and its application in the fuzzy logic control of an autonomous mobile robot”, Int. J. Adv. Robot. Syst, 10.

[33] Chaiyatham T., Ngamroo I. (2012), ”A bee colony optimization based-fuzzy logic-pid control design of electrolyzer for microgrid stabilization”, Int. J. Innov. Comput. Inf. Control, 8 (9), 6049–6066.

[34] Chaiyatham T., Ngamroo I., Pothiya S., Vachirasricirikul S. (2009),” Design of optimal fuzzy logic-PID controller using bee colony optimization for frequency control in an isolated wind-diesel system”, in: Transmission & Distribution Conference & Exposition: Asia and Pacific, IEEE, pp. 1–4.

[35] Chong C.S., Low M.Y.H., Sivakumar A.I., Gay K.L. (2006), ”A bee colony optimization algorithm to job shop scheduling simulation”, in: L.F. Perrone, F.P. Wieland, J. Liu, B.G. Lawson, D.M. Nicol, R.M. Fujimoto (Eds.), Proceedings of the Winter Conference, Washington, DC, pp. 1954–1961.

[36] Fierro R., Castillo O. (2013),” **Design of Fuzzy Different PSO Variants, Recent Advances on Hybrid Intelligent Systems** ”, Springer, pp. 81–88.

[37] Flores Mendoza J.I. , Mezura Montes E. (2008), ”Dynamic adaptation and multiobjective concepts in a particle swarm optimizer for constrained optimization”, in: CEC 2008 (IEEE World Congress on Computational Intelligence) IEEE Congress on Evolutionary Computation, 2008, IEEE, pp. 3427–3434.

[38] Hong T.P., Tung Y.F., Wang S.L., Wu M.T., Wu Y.L. (2008),” Extracting membership functions in fuzzy data mining by ant colony systems”, in: International Conference on Machine Learning and Cybernetics, vol. 7, pp. 3979–3984.

[39] Isizoh A.N., Okide S.O., Anazia A.E., Ogu C.D. (2012),” Temperature control system using fuzzy logic technique”, Int. J. Adv. Res. Artif. Intell, (IJARAI), 1 (3).

[40] Jang J.S.R., Sun C.T., Mizutani E.v, ”Neuro-fuzzy and soft computing – a computational approach to learning and machine intelligence”, IEEE Trans. Autom. Control ,42 (10), 1482–1484.

[41] Karaboga D. (2005),”An Idea based on Honey Bee Swarm for Numerical Optimization” (Technical Report-Tr06, October, 2005), Erciyes University, Engineering Faculty Computer Engineering Department, Kayseri/Türkiye.

[42] Karaboga D., Basturk B. (2008),” On the performance of Artificial Bee Colony (ABC) algorithm”, Appl, Soft Comput, 8 (2), 687–697.

[43] Karaboga D., Akay B., Ozturk C. (2007),” Artificial Bee Colony (ABC) Optimization Algorithm for Training Feed-Forward Neural Networks”, Springer Verlag, Berlin, Heidelberg , pp. 318–329.

[44] Li Z., Xu C. (2009), ”Adaptive fuzzy logic control of dynamic balance and motion for wheeled inverted pendulums”, Fuzzy Sets Syst, 160 (12) , 1787–1803.

[45] Luci P.c, Teodorovic D. (2002), ” Transportation modeling: an artificial life approach ”, in: Proceedings of the 14th IEEE, International Conference on Tools with Artificial Intelligence, Washington, DC, pp. 216–223.

[46] Luci P. Teodorovic c, D. (2003), ” Vehicle routing problem with uncertain demand at nodes: the bee system and fuzzy logic approach”, in: J.L. Verdegay (Ed.), Fuzzy Sets in Optimization, Springer-Verlag, Heidelberg, Berlin, pp. 67–82.

[47] Man K.F., Tang K.S., Kwong S. (2000), ” **Genetic Algorithms: Concepts and Designs** ”, Springer.

[48] Melin P., Olivas F., Castillo O., Valdez F., Soria J., Valdez M. (2013),”Optimal design of fuzzy classification systems using PSO with dynamic parameter adaptation through fuzzy logic”, Expert Syst. Appl, 40 (8), 3196–3206.

[49] Mendel J. (2001) , ” Uncertain Rule-Based Fuzzy Logic Systems Introduction and New Directions ”, PH PTR.

[50] Neyoy H., Castillo O., Soria J. (2013),” Dynamic fuzzy logic parameter tuning for ACO and its application in TSP problems”, in: Recent Advances on Hybrid Intelligent Systems, Berlin, Heidelberg, Springer, pp. 259–271.

[51] Qin A.K., Huang V.L., Suganthan P.N. (2009), ”Differential evolution algorithm with strategy adaptation for global numerical optimization”, IEEE Trans. Evolut. Comput, 13 (2), 398–417.

[52] Teodorovic D., DellÓrco M. (2005), ”Bee colony optimization – a cooperative learning approach to complex transportation problems”, in: Advanced OR and AI Methods in Transportation. Proceedings of the 10th Meeting of the EURO Working Group on Transportation, Poznan, Poland, pp. 51–60.

[53] Teodorovic D., Luci P. c., Markovic G., DellÓrco M. (2006),” Bee colony optimization: principles and applications”, in: B. Reljin, S. Stankovic (Eds.), Proceedings of the Eight

Seminar on Neural Network Applications in Electrical Engineering – EUREL 2006, University of Belgrade, Belgrade, pp. 151–156.

[54] Teodorovic D., Selmi c. M. (2007), ”The BCO algorithm for the p median problem”, in: of the XXXIV Serbian Operations Research Conference, Zlatibor, Serbia.

[55] Teodorovic D. (2008) ”Swarm intelligent systems for transportation engineering: principles and applications”, Transp. Res. Part C: Emerg. Technol, 16 , 651–782.

[56] Teodorovic D. (2009), ” **Bee colony optimization (BCO)** ”, in: C.P. Lim, L.C. Jain, S Dehuri (Eds.), Innovations in Swarm Intelligence, Springer-Verlag, pp. 39–60.

[57] ”The MathWorks, Fuzzy Logic Toolbox For use with Matlab”, (2006), Userś Guide Version 2.

[58] Vesterstrom J. , Thomsen R. (2004), ”A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems”, in: CEC2004 Congress on Evolutionary Computation, vol. 2, IEEE, pp. 1980–1987.

[59] Wong L.P., Low M.Y.H., Chong C.S. (2008),” Bee colony optimization with local search for traveling salesman problem”, in: Proceedings of the 6th IEEE International Conference on Industrial Informatics, pp. 1019–1025.

[60] Zadeh L. A. (1965), ”Fuzzy logic”, IEEE Comput, 83–92.

[61] Zamani A.A., Bijami E., Sheikholeslam F., Jafrasteh B. (2014),” Optimal fuzzy load frequency controller with simultaneous auto-tuned membership functions and fuzzy control rules”, Turk. J. Electr. Eng. Comput. Sci., 22 (1) , 66–86.

Aabstract

In this thesis, we study the designing of fuzzy controllers for path planning of multiple mobiling robots in dynamic unknown environments . By unknown environment, we mean the lack of a map of the entire environment, and by being dynamic; we mean the presence of moving obstacles in the environment. Our main goal is to plan the movement of a group of robots from the original dynamic mode to the final state while avoiding collisions , static and dynamic obstacles. We assume that none of the robots have complete information about the environment and do not share their knowledge with each other. So, finding the optimal route is not always possible and finding a good path (close to optimal) is valuable. In this thesis, we first formulate the problem as a linear programming. Then, using a fuzzy controller as a meta-heuristic method, we plan the routing of a robot in an unknown static environment. In the following, we extend the previous state and outline the simultaneous routing of several robots in an unknown dynamic environment using a fuzzy controller. As well, we propose an algorithm for optimizing the multi-objective robot path planning in dynamic environments, and we will show that this approach is capable of finding the near-optimal solution very fast.

keywords: path planning of multiple robots, dynamic unknown environment, fuzzy controller, optimizing the multi-objective.

Shahrood University of Technology

Faculty Of Mathematical Sciences

MSc Thesis in: Operational Research

Fuzzy-based Path Planning for Multiple Mobile Robots in Unknown Dynamic Environment

By: Zahra Barati

Supervisors

Dr. Jafar Fathali
Dr. Somaye Moghari

February 2019