



دانشگاه صنعتی امیرکبیر
(پلی‌تکنیک تهران)
دانشکده مهندسی برق

پروژه کارشناسی
گرایش کنترل

طراحی مسیر و پیاده‌سازی کنترل پلتون خودروهای
هوشمند در بستر اینترنت اشیاء

نگارش
روزبه بازرگانی

استادان راهنمای
دکتر حیدرعلی طالبی و دکتر ایمان شریفی

استاد مشاور
دکتر حیدرعلی طالبی

۱۴۰۰ فروردین

بِسْمِ اللّٰهِ الرَّحْمٰنِ الرَّحِيْمِ

صفحه فرم ارزیابی و تصویب پایان نامه- فرم تأیید اعضاء کمیته دفاع

در این صفحه فرم دفاع یا تایید و تصویب پایان نامه موسوم به فرم کمیته دفاع- موجود در پرونده آموزشی- را قرار دهید.

نکات مهم:

- نگارش پایان نامه/رساله باید به **زبان فارسی** و بر اساس آخرین نسخه دستورالعمل و راهنمای تدوین پایان نامه های دانشگاه صنعتی امیرکبیر باشد.(دستورالعمل و راهنمای حاضر)
- رنگ جلد پایان نامه/رساله چاپی کارشناسی، کارشناسی ارشد و دکترا باید به ترتیب مشکی، طوسی و سفید رنگ باشد.
- چاپ و صحافی پایان نامه/رساله بصورت **پشت و رو(دورو)** بلامانع است و انجام آن توصیه می شود.



دانشگاه صنعتی امیرکبیر
(پلی‌تکنیک تهران)

به نام خدا

تاریخ: فروردین ۱۴۰۰

تعهدنامه اصالت اثر

اینجانب روزبه بازرگانی متعهد می‌شوم که مطالب مندرج در این پایان‌نامه حاصل کار پژوهشی اینجانب تحت نظرارت و راهنمایی استاد دانشگاه صنعتی امیرکبیر بوده و به دستاوردهای دیگران که در این پژوهش از آنها استفاده شده است مطابق مقررات و روال متعارف ارجاع و در فهرست منابع و مأخذ ذکر گردیده است. این پایان‌نامه قبلاً برای احراز هیچ مدرک هم‌سطح یا بالاتر ارائه نگردیده است. در صورت اثبات تخلف در هر زمان، مدرک تحصیلی صادر شده توسط دانشگاه از درجه اعتبار ساقط بوده و دانشگاه حق پیگیری قانونی خواهد داشت.

کلیه نتایج و حقوق حاصل از این پایان‌نامه متعلق به دانشگاه صنعتی امیرکبیر می‌باشد. هرگونه استفاده از نتایج علمی و عملی، واگذاری اطلاعات به دیگران یا چاپ و تکثیر، ترجمه و اقتباس از این پایان‌نامه بدون موافقت کتبی دانشگاه صنعتی امیرکبیر ممنوع است. نقل مطالب با ذکر مأخذ بلامانع است.

روزبه بازرگانی

امضا

تَعْدِيمُهُ

مادر، مدر و برادر عزیزم

روزبه بازکانی

۱۴۰۰ فروردین

سپاس‌گزاری

حال که به پایان دوره کارشناسی رسیده‌ام وظیفه خود می‌دانم که از تمام اساتید، تدریس‌یاران، مسئولان، دوستان و دیگر کسانی که در این دوره مرا یاری کرده و موجب پیشرفت من در زمینه‌های مختلف زندگی شده‌اند، قدردانی نمایم.

در ابتدا از آقای دکتر حیدر علی طالبی که علاوه بر استاد مشاور، استاد راهنمای من در این پروژه بوده و مرا در طول این مدت بسیار راهنمایی کرده‌اند، نهایت سپاس را دارم. ضرورت آموخته‌هایم از ایشان برای اتمام این پروژه غیرقابل انکار است.

از آقای دکتر ایمان شریفی، که کلاس‌های متعدد و مفیدی را زیر نظر ایشان گذرانده‌ام و استاد راهنمای این پروژه بوده‌اند، تشکر ویژه می‌نمایم. دلسوی و مشاوره‌های ایشان در طول انجام پروژه، باعث به نتیجه رسیدن آن بوده است.

لازم می‌دانم از آقای دکتر احمد افشار که داوری این پایان‌نامه را بر عهده داشتند، سپاس‌گزاری نمایم. از خانم فیروزمند که در بخش‌های مختلف این پروژه مرا راهنمایی کرده‌اند، بسیار قدردانی می‌نمایم. از دوست عزیز خود، آقای کیان بهزاد، که با همفکری و مشاوره‌های ایشان، مرا تا پایان این پروژه همراهی کرده‌اند بسیار سپاس‌گزارم.

از آقایان، سید محمد کاشانی و حجت حسامی‌پور که پایان‌نامه‌های آن‌ها انگیزه و کمکی برای انجام این پروژه بوده است بسیار ممنون هستم.

از خانم‌ها، شاکری، رئیس‌الساداتی، مسرووری سعادت، آقایان، بهنام جباری زاده، امین رزاقی و امیر بیات و تمام دوستان دیگری که وقتی‌شان را در اختیارم قرار دادند و مرا به پایان رسانندن هر چه بهتر این پایان‌نامه کمک کرده‌اند، تشکر می‌نمایم.

در نهایت از خویشاوندان، پدر، مادر و برادرم که در تمام مراحل زندگی پشتیبان و راهنمای من بوده‌اند نهایت سپاس و ارادت را دارم.

چکیده

در این پایان‌نامه کنترل دسته‌ای از ربات‌ها توسط روش‌های سیستم‌های چندعاملی به همراه مسیریابی توسط میدان پتانسیل، الگوریتم‌های ابتکاری همانند A^* و همچنین Q-learning که یک روش یادگیری تقویتی است، شبیه‌سازی شده است. در کنترل اجماع روش‌های تک انتگرالی پیاده‌سازی گشته‌اند. به دلیل تأخیر در رسیدن اطلاعات عامل رهبر به بقیه ربات‌ها، از رهبر مجازی استفاده شده است. یک روش برای کنترل اجماع پیشنهاد گردیده و پایداری آن به کمک پایداری لیاپانوف اثبات گشته است. برای مسیریابی توسط الگوریتم Q-learning توابع پاداش مختلف بررسی و تحلیل گشته‌اند. در نهایت، دسته ربات‌ها با الگوریتم‌های مسیریابی متفاوت، در محیطی از موانع ثابت و متغیر، مورد امتحان واقع شدند.

واژه‌های کلیدی:

سیستم‌های چندعاملی، کنترل اجماع، هوش مصنوعی، روش‌های ابتکاری، یادگیری تقویتی، Q-learning، الگوریتم A^*

فهرست مطالب

عنوان

صفحه

۸	۲	مدل‌سازی و کنترل تک ربات
۹	۱-۲	مقدمه
۹	۲-۲	مدل‌سازی سینماتیکی
۱۲	۳-۲	کنترل کننده موقعیت تک ربات
۱۳	۱-۳-۲	کنترل سرعت زاویه‌ای ربات
۱۴	۲-۳-۲	کنترل سرعت خطی ربات
۱۵	۳-۳-۲	کنترل کننده خاموش روش
۱۵	۴-۲	نتایج کنترل موقعیت یک ربات
۱۹	۳	کنترل اجماع ربات‌ها
۲۰	۱-۳	مقدمه
۲۰	۲-۳	گراف اجماع
۲۱	۱-۲-۳	عدد همبندی یالی
۲۲	۲-۲-۳	کمینه‌ترین گراف اجماع
۲۴	۳-۲-۳	تشکیل گراف اجماع
۲۴	۳-۳	الگوریتم اجماع
۲۶	۱-۳-۳	خواص ماتریس لاپلاسین
۲۸	۲-۳-۳	پایداری کنترل اجماع
۳۴	۳-۳-۳	روش شکل‌دهی
۳۵	۴-۳	شبیه‌سازی و نتایج
۳۵	۱-۴-۳	شبیه‌سازی اجماع عددی
۳۶	۲-۴-۳	شبیه‌سازی با ربات‌ها
۳۹	۴	مسیریابی
۴۰	۱-۴	مقدمه
۴۰	۲-۴	میدان پتانسیل

۴۰	۱-۲-۴ روابط میدان پتانسیل
۴۲	۲-۲-۴ کنترل مسیر ربات با میدان پتانسیل
۴۳	۳-۴ الگوریتم‌های ابتکاری
۴۴	۱-۳-۴ الگوریتم <i>BFS</i>
۴۶	۲-۳-۴ الگوریتم Greedy best-first search
۴۷	۳-۳-۴ الگوریتم <i>A*</i>
۴۸	۴-۳-۴ نتایج و مقایسه الگوریتم‌های ابتکاری
۵۲	۴-۴ Q-learning
۵۳	۱-۴-۴ الگوریتم Q-learning
۵۴	۲-۴-۴ پیاده‌سازی الگوریتم Q-learning
۵۴	۳-۴-۴ نتایج مسیریابی با Q-learning
۵۷	۴-۴-۴ اشتباهات رایج در انتخاب پاداش
۵۹	۵-۴-۴ حل مثال با الگوریتم Q-learning
۶۲	۶-۴-۴ مقایسه زمانی الگوریتم Q-learning با الگوریتم‌های ابتکاری
۶۴	۵ مسیریابی برای اجماع ربات‌ها
۶۵	۱-۵ مقدمه
۶۵	۲-۵ میدان پتانسیل
۶۵	۱-۲-۵ جلوگیری از برخورد با موانع با شکل‌گیری ثابت
۶۸	۲-۲-۵ جلوگیری از برخورد با موانع با شکل‌گیری شکننده
۶۹	۳-۵ روش‌های ابتکاری
۶۹	۱-۳-۵ <i>BFS</i>
۷۰	۲-۳-۵ Greedy best-first search
۷۱	۳-۳-۵ <i>A*</i>
۷۲	۴-۵ Q-learning
۷۴	۵-۵ مسیریابی در حضور موانع متحرک
۷۸	۶ جمع‌بندی و نتیجه‌گیری و پیشنهادات
۷۹	۱-۶ نتیجه‌گیری

فهرست مطالب

۸۰	۲-۶ پیشنهادات
۸۲	منابع و مراجع

فهرست اشکال

شکل

صفحه

۱-۱	شماي کلي از نحوه کار سیستم‌های دسته‌ای	۵
۱-۲	ربات استفاده شده به همراه پارامترهای آن	۱۰
۲-۲	دایره به همراه قطاع و کمان آن	۱۰
۳-۲	چرخش ماشین حول یک نقطه مشخص	۱۱
۴-۲	دامنه و برد تابع استفاده شده	۱۳
۵-۲	کنترل‌کننده تک ربات	۱۶
۶-۲	موقعیت مطلوب و موقعیت واقعی ربات	۱۶
۷-۲	موقعیت واقعی و موقعیت فیدبک گرفته شده از سنسور ربات	۱۷
۸-۲	سرعت خطی ربات	۱۷
۹-۲	سرعت زاویه‌ای چرخ‌های ربات	۱۸
۱۰-۲	مسیر پیموده شده توسط ربات در زاویه‌ای نزدیک به π	۱۸
۱-۳	بردار ویژه متناظر با مقدار ویژه صفر در ماتریس لاپلاسین گراف دارای k مولفه همبند	۲۷
۲-۳	بلوک‌های شبیه‌سازی شده برای اجماع عددی	۳۵
۳-۳	موقعیت مکان‌های شبیه‌سازی شده	۳۶
۴-۳	بلوک‌های شبیه‌سازی شده برای اجماع ربات‌ها با رهبر واقعی	۳۶
۵-۳	موقعیت مکان‌های ربات‌ها برای اجماع با رهبر واقعی	۳۷
۶-۳	بلوک‌های شبیه‌سازی شده برای اجماع ربات‌ها با رهبر مجازی	۳۸
۷-۳	موقعیت مکان‌های ربات‌ها برای اجماع با رهبر مجازی	۳۸
۱-۴	مسیر پیموده شده توسط ربات با میدان پتانسیل	۴۲
۲-۴	گیر کردن ربات در کمینه محلی با میدان پتانسیل	۴۳
۳-۴	شبه کد الگوریتم <i>BFS</i>	۴۵
۴-۴	نحوه اعمال شدن الگوریتم <i>BFS</i>	۴۶
۵-۴	نقشه‌ای که الگوریتم‌ها به آن اعمال شده‌اند	۴۸
۶-۴	مسیر پیشنهادی توسط الگوریتم <i>BFS</i>	۴۹

۵۰	۷-۴ مسیر پیشنهادی توسط الگوریتم Greedy best-first search
۵۱	۸-۴ مسیر پیشنهادی توسط الگوریتم A^*
۵۲	۹-۴ مقایسه زمانی سه الگوریتم BFS، Greedy best-first search و A^*
۵۵	۱۰-۴ نتیجه الگوریتم Q-learning برای یادگیری تصادفی
۵۶	۱۱-۴ نتیجه الگوریتم Q-learning برای یادگیری با $R = -1$
۵۷	۱۲-۴ نتیجه الگوریتم Q-learning برای یادگیری با $R = -\ a\ _2$
۵۸	۱۳-۴ نتیجه الگوریتم Q-learning برای یادگیری با $R = -\frac{\ (y-goal)\ _2}{1000}$
۵۹	۱۴-۴ نتیجه الگوریتم Q-learning برای یادگیری با $R = -\frac{\ (y-goal)\ _2 - \ (x-goal)\ _2}{1000}$
۶۰	۱۵-۴ نقشه ورودی داده شده به الگوریتم Q-learning
۶۰	۱۶-۴ زمان طول کشیده برای یافتن جواب توسط Q-learning و مقدار ماکریم تغییر در هر گام
۶۱	۱۷-۴ مقدار ماتریس Q پس از پایان الگوریتم Q-learning
۶۱	۱۸-۴ مقدار ماتریس V پس از پایان الگوریتم Q-learning
۶۲	۱۹-۴ سیاست نهایی برای هر حالت در پایان الگوریتم Q-learning
۶۲	۲۰-۴ مسیر پیشنهادی توسط الگوریتم Q-learning
۶۳	۲۱-۴ مقایسه زمانی الگوریتم Q-learning با الگوریتم‌های ابتکاری
۶۴	۱-۵ بلوک‌های شبیه‌سازی در روش شکل‌گیری ثابت با محاسبات متمنکز
۶۶	۲-۵ مکان ربات‌ها در روش شکل‌گیری ثابت با محاسبات متمنکز
۶۷	۳-۵ بلوک‌های شبیه‌سازی در روش شکل‌گیری ثابت با محاسبات نامترنکز
۶۷	۴-۵ مکان ربات‌ها در روش شکل‌گیری ثابت با محاسبات نامترنکز
۶۸	۵-۵ بلوک‌های شبیه‌سازی در روش شکل‌گیری شکننده
۶۸	۶-۵ مکان ربات‌ها در روش شکل‌گیری شکننده
۶۹	۷-۵ بلوک‌های شبیه‌سازی در روش BFS
۷۰	۸-۵ مکان ربات‌ها در روش BFS
۷۰	۹-۵ بلوک‌های شبیه‌سازی در روش Greedy best-first search
۷۱	۱۰-۵ مکان ربات‌ها در روش Greedy best-first search
۷۱	۱۱-۵ بلوک‌های شبیه‌سازی در روش A^*

۷۲	۱۲-۵ مکان ربات‌ها در روش A^*
۷۳	۱۳-۵ بلوک‌های شبیه‌سازی در روش Q-learning
۷۳	۱۴-۵ مکان ربات‌ها در روش Q-learning
۷۴	۱۵-۵ بلوک‌های شبیه‌سازی در روش Q-learning همراه با مانع متحرک
۷۵	۱۶-۵ مکان ربات‌ها در روش Q-learning همراه با مانع متحرک
۷۵	۱۷-۵ مکان ربات‌ها در روش میدان پتانسیل همراه با مانع متحرک
۷۶	۱۸-۵ مکان ربات‌ها در روش BFS همراه با مانع متحرک
۷۶	۱۹-۵ مکان ربات‌ها در روش Greedy best-first search همراه با مانع متحرک
۷۷	۲۰-۵ مکان ربات‌ها در روش A^* همراه با مانع متحرک

فهرست جداول

صفحه

جدول

فصل اول

مقدمه

۱-۱ مقدمه

در این فصل ابتدا به بیان هدف پروژه و انگیزه از طرح آن پرداخته می‌شود. سپس دلیل نیاز و فواید سیستم‌های چندعاملی پرداخته خواهد شد. در ادامه به تاثیر هوش مصنوعی در رباتیک اشاره خواهد شد و در انتهای شما کلی از فصل‌های پایان‌نامه ارائه می‌شوند.

۲-۱ هدف و انگیزه

امروزه با توسعه هوشمندسازی، انواع مختلف ربات‌ها با توانمندی‌های متنوع مورد استفاده قرار می‌گیرند. عملکرد گروه ربات‌ها و تعامل بین آن‌ها در صنایع مختلف کاربرد دارد. که یکی از انواع آن، سیستم چند عاملی^۱ و به طور خاص اجماع^۲ است. منظور از اجماع دسته‌ای از ربات‌های است که به هدایت یکی از آن‌ها، در کنار یکدیگر با نظم و ترتیب، فاصله استاندارد و با سرعت بهینه در کنار یکدیگر حرکت کنند.

در این پروژه هدف کنترل ربات‌های چرخ‌دار متوجه^۳، مدل دیفرانسیلی^۴، نیمه خودران و پیاده‌سازی اجماع ربات‌ها است که این سیستم به کمک شبیه‌ساز^۵ متلب^۶ و برای نمایش به کمک ابزار شبیه‌سازی ربات‌های متوجه^۷ پیاده سازی گردیده است.

با بهره‌گیری از فناوری اینترنت اشیاء^۸ می‌توان ارتباط ربات‌ها را در بستر اینترنت به یکدیگر و یا به مراکز مختلف میسر نمود. به این ترتیب می‌توان به کمک آن برای ربات‌ها مسیر طراحی و ترافیک مسیر و اجماع ربات‌ها را تا رسیدن به مقصد کنترل کرد.

از مزایای اجماع عبارت از تسريع در عملیات و کاهش انرژی مصرفی و استهلاک ربات‌ها است. از جمله کاربردهای دسته‌ای می‌توان به خودروهای سنگین نیمه هوشمند بین شهری، در جاده‌های اختصاصی شرکت‌ها برای جابه‌جایی مواد اولیه و فراورده‌ها و درون انبارها اشاره کرد. همچنین در جابه‌جایی اجسامی که یک ربات به تنها‌یی از انجام آن ناتوان است، در عملیات‌های امدادی جست و جو و حتی در فضادسته‌ای استفاده می‌گردد که نشان از اهمیت بسیار بالای آن است.

¹multi-agent systems

²platoon

³mobile wheeled robots

⁴differential model

⁵Simulink

⁶MATLAB

⁷mobile robotics simulation toolbox

⁸Internet of Things

امروزه علم مدیریت ترافیک با ترکیب تکنولوژی‌های پیشرفته با زیر ساخت‌های شهری، بسیار نوین گشته است. این امر باعث جذب علاقه مدیران برای استفاده از سازمان حمل و نقل هوشمند^۹ شده است [۱]. رکن اصلی سازمان حمل و نقل هوشمند خودرو هوشمند^{۱۰} است. خودروهای هوشمند دارای ویژگی‌هایی هستند که از مهمترین آن‌ها می‌توان به خودران^{۱۱} بودن خودروها اشاره کرد. خودران بودن خودروها علاوه بر اینکه راحتی وقت بیشتر به خاطر آزاد بودن سرنشینان را به همراه دارد، می‌تواند از اشتباه‌های رانندگان خودروها نیز جلوگیری و به راحتی مدیریت کرد. در نتیجه ترافیک کمتر و امنیت بیشتری را در سطح جاده‌ها فراهم آورد.

ایده جذاب دیگر این است که به جای کنترل یک خودرو، گروهی از خودروها را کنترل کرد. به این ترتیب که هر خودرو وظیفه مسیریابی نداشته باشد و بتوان الگوریتم‌های مسیریابی را بهینه کرد. برای مثال می‌توان فاصله خودروها را به نسبت سرعتشان کمتر کرد تا از ظرفیت جاده‌ها بیشتر استفاده گردد. موضوع مهم دیگر رابطه هوش مصنوعی^{۱۲} و رباتیک است. رباتیک بر پایه پیشرفتهای ایجاد شده در مکاترونیک، مهندسی برق و محاسبه، عملکرد موتورهای سنسورداری را که قابلیت وفق دادن خود با محیط همیشه در حال تغییر را دارند، ارتقا می‌دهد. تا به حال، سیستم‌های صنعتی، به صورت محیط مناسب برای توانایی‌ها و ویژگی‌های ربات ساخته می‌شد اما امروزه ربات‌ها می‌توانند در محیط‌های مختلفی خود را وفق دهند. اتوماسیون ربات، می‌تواند به سه دسته ادراک^{۱۳}، برنامه‌ریزی^{۱۴} و اجرا^{۱۵} (اثرگزاری^{۱۶}، پیمایش^{۱۷} و همکاری^{۱۸}) تقسیم شود. هدف از ادغام هوش مصنوعی و رباتیک، بهینه‌کردن اتوماسیون^{۱۹} با یادگیری است. این سطح از هوش را می‌توان با توانایی پیش‌بینی آینده، چه در برنامه‌ریزی یک عمل یا تعامل با محیط، اندازه‌گیری کرد. با وجود اینکه ساخت سیستمی که همانند انسان هوشمند باشد، هنوز موفقیت‌آمیز نبوده است، ربات‌ها امروزه قادر به انجام کارهای بسیار تخصصی‌ای مثل راندن خودرو، پرواز در محیط‌های طبیعی و ساخته انسان، شنا، حمل جعبه و اجسام در ناحیه‌های مختلف، برداشت و قرار دادن احسام، هستند [۹].

⁹Intelligent Transportation System

¹⁰intelligent vehicle

¹¹autonomous vehicles

¹²artificial intelligence

¹³perceiving

¹⁴planning

¹⁵execution

¹⁶manipulating

¹⁷navigating

¹⁸collaborating

¹⁹automation

در این پایان‌نامه الگوریتم‌های هوش مصنوعی برای مسیریابی دسته‌ی ربات‌ها مورد بررسی و مقایسه قرار گرفته‌اند تا بهترین روش در این بین برای مسیریابی ربات‌ها، از نظر زمانی، مسافت و امنیت یافتد.^{۲۰}

۳-۱ کارهای پیشین

امروزه تلاش‌ها و تحقیقات بسیاری در زمینه خودروهای هوشمند و بهره بردن از فناوری‌های مختلف در این زمینه در حال انجام است. به عنوان مثال شرکت تسلا^{۲۱} با ساخت ماشین‌های الکتریکی خودران با توانایی ارتباط با مرکزهای کنترل ترافیک، توانسته به یکی از غول‌های این صنعت تبدیل گردد. در مراکز علمی نیز با کمک ربات‌ها، شبیه‌سازی و سایر نمونه‌ها، سعی در ارتقا روش‌های مورد استفاده در خودروهای هوشمند را دارند. یکی از موضوعات بسیار مورد توجه همکاری ربات‌ها به عنوان یک سیستم به جای انفرادی عمل کردن آن‌ها بوده است. به عنوان مثال به جای اینکه هر ربات به تنها یی مسیریابی را انجام دهد، ربات‌ها به صورت یک دسته با یکدیگر همکاری و انتقال اطلاعات داشته باشند تا مسیریابی انجام گردد. این ایده‌ای است که برای کنترل خودروها می‌توان استفاده کرد تا بتوان از ظرفیت جاده‌ها به بیشترین حد ممکن بهره برد و مشکل ترافیک را حل نمود.

با توجه به این مسئله برای دسته‌ای از خودروها، در کنترل بخشی به نام کنترل دسته‌ای وجود دارد. در این بخش برای کنترل دسته‌ای خودروها از تئوری‌های سیستم‌های چند عاملی استفاده می‌شود. در این مقوله می‌توان مطالبی که در سیستم‌های چند عاملی مورد بررسی است را به ۴ دسته تقسیم نمود.^{۲۲}

- مدل‌سازی هر گره^{۲۳}

- نحوه انتقال داده بین گره‌ها^{۲۴}

- قرارگیری ربات‌ها

- کنترل توزیع شده

²⁰Tesla

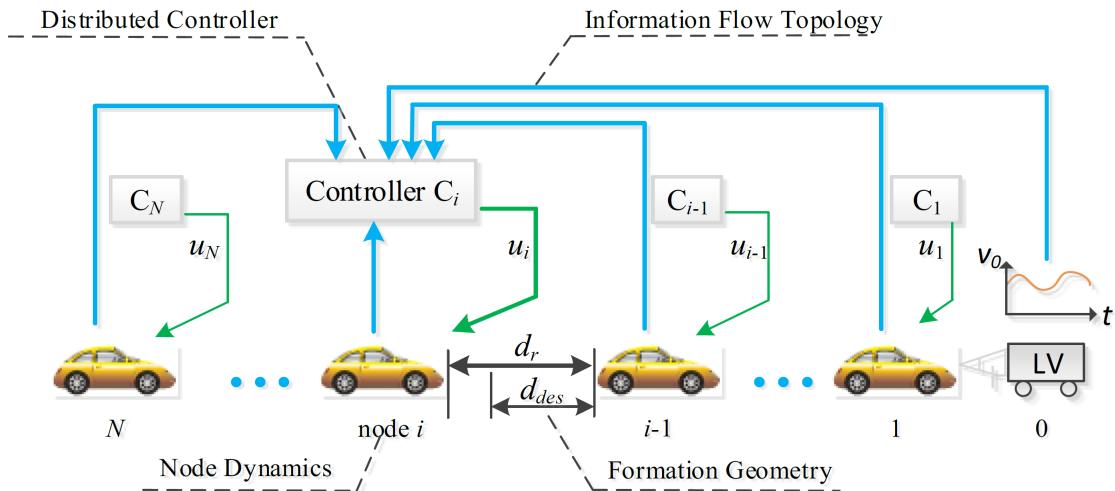
²¹node dynamic

²²information flow graph

²³formation geometry

²⁴distributed control

در شکل زیر شمای کلی یک دسته از ماشین‌ها را نشان می‌دهد که در حال حرکت به دنبال هم هستند.



شکل ۱-۱: شمای کلی از نحوه کار سیستم‌های دسته‌ای

همانطور که در شکل ۱-۱ مشاهده می‌شود هر ماشین باید به اندازه کافی هوشمند باشد تا بتواند موقعیت خود را بدست آورده و حداقل به یک یا تعداد بیشتری از خودروها ارسال کند به طوری که به طور غیر مستقیم همگی به یکدیگر مرتبط باشند. سپس با توجه به موقعیت ارسالی از هر ماشین به کمک یک کنترل‌کننده که در هر ماشین قرار دارد به موقعیت مطلوب برسد تا بتوانند با هم در مسیر مطلوب به شکل یک دسته حرکت نمایند.

در این مقوله برای استفاده بهینه از این خودروهای خودران در جاده‌ها به کنترل دسته‌ای آن‌ها مانند شکل ۱-۱ روی می‌آورند به طوری که در این دسته یک رهبر^{۲۵} و بقیه ماشین‌ها دنبال‌کننده‌ی^{۲۶} رهبر باشند.

علاوه بر این موضوع، برنامه‌ریزی برای رسیدن به بهترین مسیر نیز یکی دیگر از نکات مهم برای حمل و نقل و خودروهای هوشمند است. منظور از برنامه‌ریزی، یافتن یک طرح معقول برای اجرای آن است. برای مسیریابی روش‌های مختلفی همانند میدان پتانسیل^{۲۷}، روش نمونه‌گیری^{۲۸}، کنترل پیش‌بین^{۲۹}

²⁵leader

²⁶follower

²⁷potential field

²⁸sampling based method

²⁹model predictive control

روش‌های احتمالاتی^{۳۰} [۷] و یادگیری^{۳۱} وجود دارند که هر کدام به طور کامل در فصل ۴ به طور کامل توضیح داده شده‌اند.

۴-۱ طرح مسئله

برای پیاده‌سازی مسئله طرح شده با هدف کنترل اجماع و مسیریابی ربات‌ها، همانطور که گفته شد از برنامه MATLAB و mobile robotics simulation toolbox استفاده شده است. برای تست‌های اولیه نتایج کنترل تک ربات و در انتها با ۳ و ۴ ربات پلتون و مسیریابی آن انجام شده است. مدل ربات‌ها به صورت دیفرانسیلی^{۳۲} در نظر گرفته شده است که به معادلات آن در فصل ۲ پرداخته شده است. اطلاعات ما از محیط از سوی دیگاه پرنده^{۳۳} از محیط گرفته می‌شود. مسیریابی‌ها قابل برنامه‌ریزی به صورت آفلاین، با اجرای فایل m. هر الگوریتم و سپس اجرا شبیه‌سازی مربوطه و همینطور به صورت آنلاین قابل برنامه‌ریزی است. سعی شده است که انواع روش‌های مسیریابی پیاده‌سازی و مورد بررسی قرار گیرند. همچنین تا سطح موانع متحرک، مسیریابی و کنترل اجماع به صورت بلاذرنگ^{۳۴} ارتقا یافته است. روش‌های مسیریابی استفاده شده به شرح زیر هستند:

۱. میدان پتانسیل

۲. روش‌های نمونه‌برداری و ابتکاری

^{۳۴}BFS •

Greedy best-first search •

A^* •

۳. روش‌های یادگیری تقویتی^{۳۵}

Q-learning •

³⁰learning

³¹differential

³²bird's eye view

³³real time

³⁴Breadth first search

³⁵Reinforcement learning

پیاده‌سازی این روش‌ها دارای مهارت و دانش در مورد هوش مصنوعی و روش تبدیل کردن مسئله مسیریابی برای ربات‌ها به یک مسئله دارای حالت^{۳۶}، اقدامات^{۳۷} و تبادل اطلاعات بین ربات و محیط است که در فصل ۴ به طور مفصل در مورد آن‌ها صحبت شده است.

نکته قابل توجه، ساده‌نویسی کد است به طوری که به راحتی بتوان آن را با زبان C نیز درک و نوشت تا بتواند بر روی دو ربات مستقر در آزمایشگاه بلادرنگ که مدل آن‌ها نیز دیفرانسیلی است، پیاده‌سازی کرد.

۱-۵ قالب‌بندی پایان‌نامه

این پایان‌نامه شامل ۴ فصل اصلی به شرح زیر است:

۱. مدل‌سازی و کنترل تک ربات

۲. کنترل اجماع ربات‌ها

۳. مسیریابی

۴. مسیریابی برای اجماع ربات‌ها

در فصل دوم، پس از به دست آوردن روابط ربات و مدل‌سازی به کنترل موقعیت آن انجام می‌شود. در فصل سوم، پس از گذری بر مفاهیم نظریه گراف و خواص ماتریس لالپاسین، الگوریتم اجماع توضیح داده شده و پایداری آن اثبات می‌گردد. در فصل چهارم روش‌های مسیریابی گوناگونی پیاده‌سازی شده‌اند و همچنین محیط به نحوی که بتوان الگوریتم‌هایی که با حالت و اقدامات تعریف می‌شوند، به خصوص Q-learning، را پیاده‌سازی کرد، حالت‌بندی گشته است. در فصل پنجم نیز، نتایج مسیریابی و کنترل برای اجماع ربات‌ها با موانع ثابت و متحرک بررسی گشته است.

³⁶state

³⁷actions

فصل دوم

مدل‌سازی و کنترل تک ربات

۱-۲ مقدمه

در این فصل چگونگی کنترل موقعیت برای هر یک از ربات‌ها توضیح داده شده است. در این پژوهش از ربات‌های چرخدار دیفرانسیلی^۱ استفاده شده است که ابتدا باید مدل ربات را به دست آورد. در این پژوهش از مدل سینماتیکی^۲ ربات استفاده شده و وارد مدل دینامیکی نشده‌ایم. پس از توضیح مدل سینماتیکی، بر روی آن کنترل موقعیت پیاده سازی می‌شود. با توجه به مدلی که به دست آمده، می‌توان دریافت که متغیرهای کنترلی، سرعت چرخ‌های چپ و راست هستند و با کنترل آن‌ها و ادغام آن با سینماتیک معکوس، می‌توان کنترل موقعیت را انجام داد.

کنترل تک ربات دارای اهمیت بسیار زیادی است زیرا درون حلقه کنترل اجماع قرار می‌گیرد. حلقه داخلی باید ۵ تا ۱۰ برابر سریع‌تر از حلقه خارجی باشد تا کنترل کننده خارجی دچار مشکل نشود، در نتیجه سرعت پاسخ این کنترل کننده برای هدف نهایی بسیار اهمیت دارد. حال که از اهمیت بالای این فصل آگاه شدیم، به بررسی نحوه بدست آوردن مدل سینماتیکی ربات‌ها می‌پردازیم.

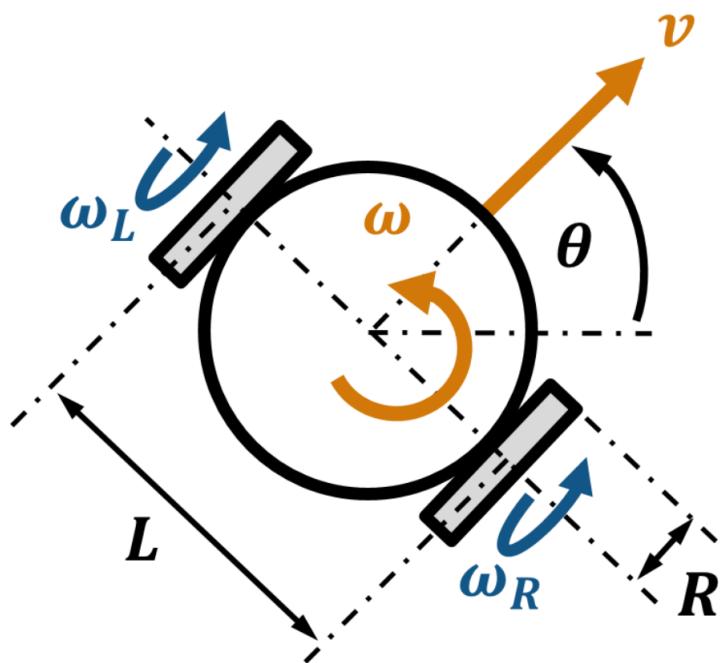
۲-۲ مدل‌سازی سینماتیکی

در شکل ۱-۲ عکسی از ربات چرخدار دیفرانسیلی مشاهده می‌کنیم. ورودی این ربات ω_L و ω_R که سرعت زاویه‌ای چرخ چپ و راست است، می‌باشد. همچنین خروجی ربات، سرعت خطی ربات، v ، و سرعت زاویه‌ای ربات، ω است. شعاع چرخ، R ، و فاصله بین دو چرخ، L ، مشخصات فیزیکی ربات هستند. در نهایت موقعیت ربات توسط مکان مرکز آن در بیان صفحه مختصات $x-y$ و همچنین اختلاف زاویه رو به روی ربات با محور x ‌ها، با θ بیان می‌شود. با توجه به شکل ۲-۲ اگر زاویه قطاع را θ بگیریم، داریم:

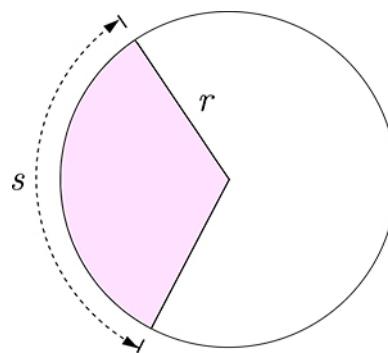
$$s = r\theta \quad (1-2)$$

¹Differential wheeled robot

²Kinematic model



شکل ۲-۱: ربات استفاده شده به همراه پارامترهای آن



شکل ۲-۲: دایره به همراه قطاع و کمان آن

پس اگر چرخ را کاملاً دایره فرض کنیم و طول کمان را همچنان s بگیریم، خواهیم داشت:

$$s = R\theta \quad (2-2)$$

سپس با صرف نظر از تغییرات شعاع چرخ، به کمک یک مشتق به رابطه بین سرعت خطی و سرعت زاویه ای برای هر چرخ می‌رسیم:

$$v = \dot{s} = R\dot{\theta} = R\omega \quad (3-2)$$

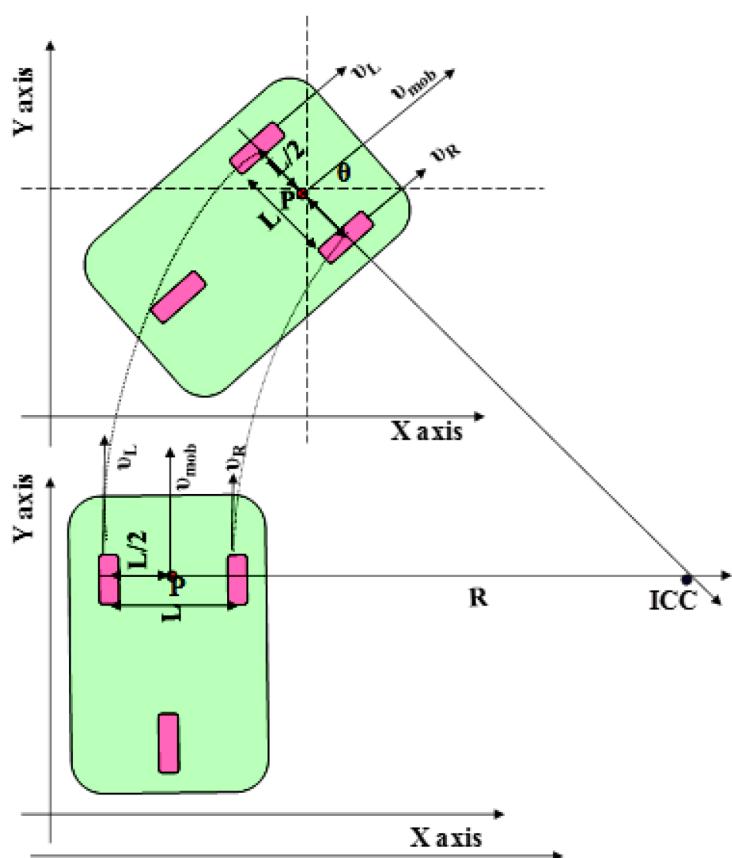
سرعت خطی ربات با میانگین سرعت خطی هر چرخ برابر است. پس لگر سرعت خطی چرخ راست و چپ را به ترتیب با v_L و v_R نشان دهیم، داریم:

$$v = \frac{v_R + v_L}{2} \quad (4-2)$$

با ادغام روابط ۳-۲ و ۴-۲ داریم:

$$v = \frac{R}{2}(\omega_R + \omega_L) \quad (5-2)$$

برای محاسبه سرعت زاویه‌ای ربات از **۴-۲** استفاده می‌شود که ربات همانند شکل ۳-۲ حول آن دوران می‌کند. قابل توجه است که چون دو چرخ آخر خودرو آزادی عمل ندارند، می‌توان خودرو را با ربات چرخدار دیفرانسیلی مدل کرد.



شکل ۲-۳: چرخش ماشین حول یک نقطه مشخص

³Instantaneous center of curvature

مقدار ICC طبق رابطه زیر به دست می‌آید [۱۱]:

$$ICC = (x - R\sin(\theta), y + R\cos(\theta)) \quad (6-2)$$

که R فاصله بین مرکز دو چرخ جلوی خودرو و نقطه‌ای است که حول آن دوران می‌کند. پس داریم:

$$R = \frac{L}{2} \times \frac{v_L + v_R}{v_L - v_R} \quad (7-2)$$

حال اگر دایره را به مرکز ICC و شعاع R در نظر بگیریم، با توجه به روابط ۴-۲، ۳-۲ و ۷-۲ داریم:

$$\omega = \frac{v}{R} = \frac{R}{L}(v_R - v_L) \quad (8-2)$$

در نتیجه با توجه به معادلات ۵-۲ و ۸-۲، سینماتیک معکوس ربات برابر است با:

$$\omega_L = \frac{1}{R}\left(v - \frac{\omega L}{2}\right), \quad \omega_R = \frac{1}{R}\left(v + \frac{\omega L}{2}\right) \quad (9-2)$$

حال اگر در کنترل کننده موقعیت، مقدار v و ω را محاسبه کرد، می‌توان به کمک رابطه ۹-۲ سرعت زاویه‌ای چرخ‌ها را طوری تنظیم نمود که ربات به سرعت خطی و زاویه‌ای ایده‌آل خود برسد.

۳-۲ کنترل کننده موقعیت تک ربات

در این قسمت هدف این است که موقعیت ربات، یا به عبارتی مختصات مرکز ربات را در صفحه $x-y$ کنترل کرد. متغیر کنترلی، سرعت زاویه‌ای چرخ‌ها می‌باشد. حال هدف این است که با محاسبه خطای و به کمک کنترل کننده، خروجی کنترل کننده را به سرعت خطی و زاویه‌ای ربات تبدیل کنیم تا در نهایت با کمک از رابطه سینماتیک معکوس ۹-۲ بتوان به سرعت زاویه‌ای چرخ‌ها رسید.

۱-۳-۲ کنترل سرعت زاویه‌ای ربات

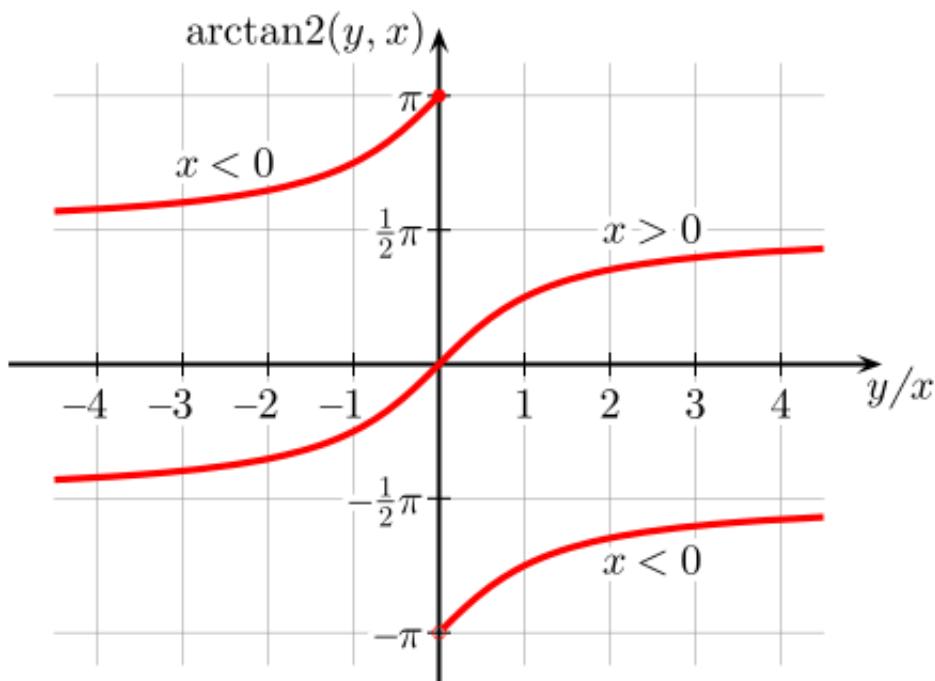
در این کنترل کننده هدف ما این است که سر ربات به سمت موقعیت مرجع باشد. در نتیجه پس از محاسبه خطای داریم:

$$e = \begin{bmatrix} x_d - x \\ y_d - y \end{bmatrix} \quad (10-2)$$

که در آن x_d و y_d موقعیت مرجع و x و y موقعیت فعلی ربات هستند.

حال با توجه به خطای به دست آمده، با کمک از تابع Atan2 زاویه‌ای که ربات باید در آن جهت قرار بگیرد، محاسبه می‌گردد. مزیت استفاده از این تابع این است که برد آن در بازه $-\pi$ تا π می‌باشد و باعث می‌شود که جهت روبروی ربات اهمیت پیدا کند. قابل ذکر است که اگر از arctan استفاده می‌شود، آنگاه فقط راستای روبروی ربات اهمیت پیدا می‌کرد و برای حالتی که ربات رو به نقطه مرجع یا پشت به آن می‌باشد، فرقی قائل نمی‌شود.

در شکل ۴-۲ دامنه و برد تابع Atan2 قابل مشاهده است.



شکل ۴-۲: دامنه و برد تابع استفاده شده

اکنون با کمک این تابع زاویه‌ای که ربات باید در آن جهت قرار گیرد، θ_d ، به صورت زیر محاسبه

می‌گردد:

$$\theta_d = \text{Atan2}(y_d - y, x_d - x) \quad (11-2)$$

که ورودی‌های تابع، عناصر بردار خطأ هستند.

حال که زاویه مرجع به دست آمد، با کم کردن زاویه حال حاضر ربات، به خطای زاویه می‌رسیم. سپس با استفاده از کنترل کننده انگرال‌گیر نسبت، به سرعت زاویه‌ای ربات دست پیدا می‌کنیم. در نتیجه با تنظیم سرعت زاویه‌ای چرخ‌ها به طوری که به سرعت زاویه‌ای مد نظر در ربات برسیم، می‌توان چهت‌گیری ربات را کنترل نمود. تنها یک مشکل بسیار مهم باقی مانده است. تغییرات زاویه پیوسته نیست و بین مقادیر π و $-\pi$ جهش وجود دارد. برای رفع این مشکل از روابط زیر استفاده شده است. از پیش می‌دانیم که مقادیر زاویه‌ای سنسورها و مطلوب در بازه $[-\pi, \pi]$ قرار دارند.

$$\delta\theta = \theta_d - \theta$$

$$sign = sgn(\delta\theta)$$

$$\bar{\delta\theta} = \delta\theta - 360 * sign$$

$$\delta\theta_{new} = smaller(\delta\theta, \bar{\delta\theta})$$

منظور از $smaller$ انتخاب مقداری است که اندازه‌اش از دیگری کوچکتر باشد. در قسمت بعد، به چگونگی کنترل سرعت خطی ربات پرداخته خواهد شد.

۲-۳-۲ کنترل سرعت خطی ربات

برای کنترل سرعت خطی ربات، فاصله بین مکان حال تا موقعيت هدف، به عنوان ورودی کنترلی مد نظر قرار می‌گیرد. قابل ذکر است که این فاصله برابر با نرم دو بردار اختلاف این دو موقعیت است و از این رابطه در کنترل استفاده شده است. در نتیجه ورودی کنترلی برابر است با:

$$\|e\|_2 = \sqrt{(x_d - x)^2 + (y_d - y)^2} \quad (12-2)$$

در این قسمت از کنترل کننده نسبت مشتق‌گیر^۴ استفاده شده است. دلیل استفاده نیز این بود که سرعت ربات دارای اهمیت است، در نتیجه از انتگرال‌گیر که سرعت را کاهش می‌دهد، استفاده نشده است. از طرفی چون پاسخ گذرا هنگام کنترل سرعت دارای اهمیت است، از مشتق‌گیر استفاده شده است تا پاسخ را بهبود بخشد. همچنین برای جلوگیری از وارد شدن ولتاژ بیش از حد به ربات، در خروجی کنترلی اشباع قرار داده شده است تا سرعتی که از ربات می‌خواهیم و در نتیجه ولتاژ اعمالی به موتور چرخ‌ها، محدود گردد.

۳-۳-۲ کنترل کننده خاموش روشن

برای کاهش تاثیر نویز، قبل از کنترل کننده سرعت و زاویه، از کنترل کننده خاموش روشن با بازه $0/3^{\circ}$ استفاده شده است. علاوه بر این در کنترل کننده سرعت از مقدار خروجی در حالت قبل نیز فیدبک گرفته شده است. دلیل این کار این است که ربات در حدود فاصله تعیین شده می‌ایستاد و نسبت به نویز با مقدار بسیار کم نیز حساس می‌ماند. با بررسی خروجی گذشته با اینکه به مقدار کمتر از $0/05^{\circ}$ رسیده باشد، خیال‌مان راحت خواهد شد زمانی سرعت صفر خواهد شد، که بسیار به موقعیت مطلوب نزدیک شده باشیم. و در نتیجه به نویز در حدود $0/3^{\circ}$ نیز مقاوم خواهیم بود. حال نتایج آزمایشات برای یک ربات، مورد بررسی قرار می‌گیرد.

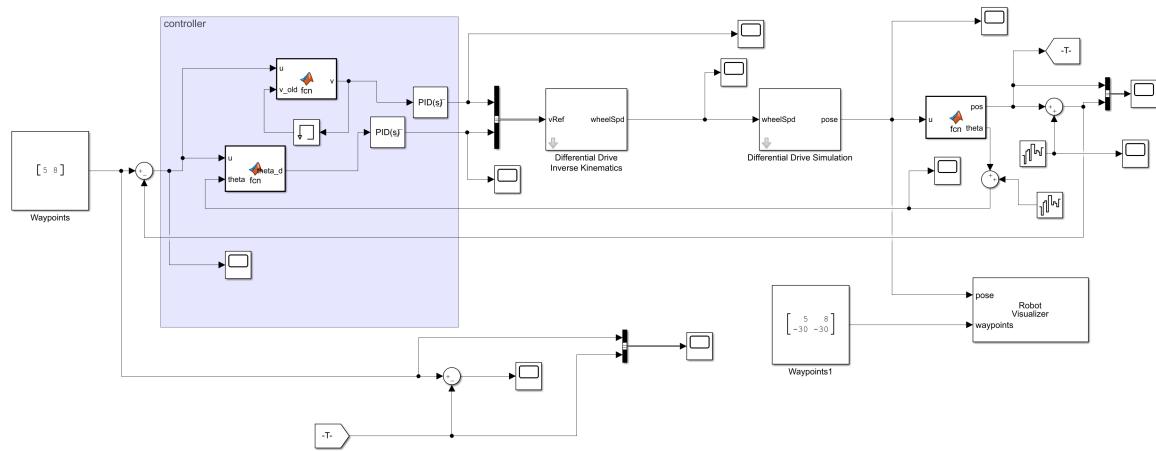
۴-۲ نتایج کنترل موقعیت یک ربات

کنترل کننده تک ربات در شکل ۵-۲ آمده است. در قسمت بنفسرنگ بلوک‌های کنترلی به طور خاص مشخص شده‌اند.

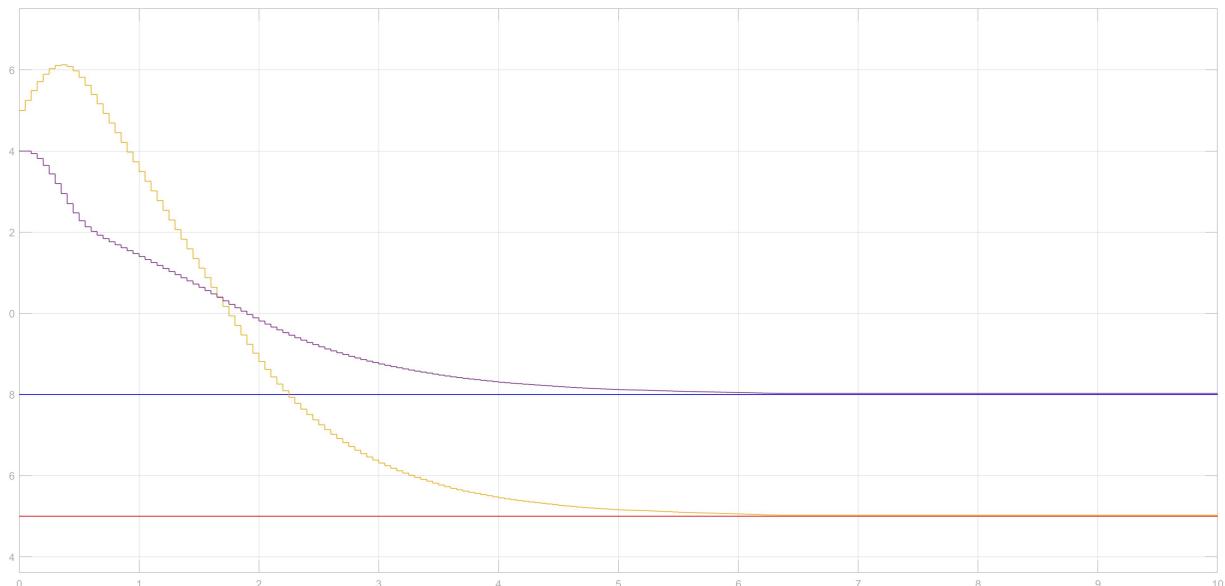
در کنترل سرعت، کنترل کننده نسبت با $P = 1$ پیاده‌سازی شده است. در کنترل کننده زاویه، کنترل کننده نسبت با $P = 10$ برای سرعت بیشتر به دلیل اینکه اول باید زاویه درست شود استفاده شده است.

نویز استفاده شده به صورت تابع احتمال گاوی، و با بیشترین خطای $1/0$ متر به سنسورها اعمال شده است. موقعیت اولیه برابر $[14/15]^T$ و موقعیت مطلوب برابر $[8/5]^T$ بوده است. در نمودار ۶-۲ موقعیت واقعی ربات و موقعیت مطلوب، نمایش داده شده است. که نشان از تعقیب مناسب نقطه مطلوب توسط ربات است.

⁴PD

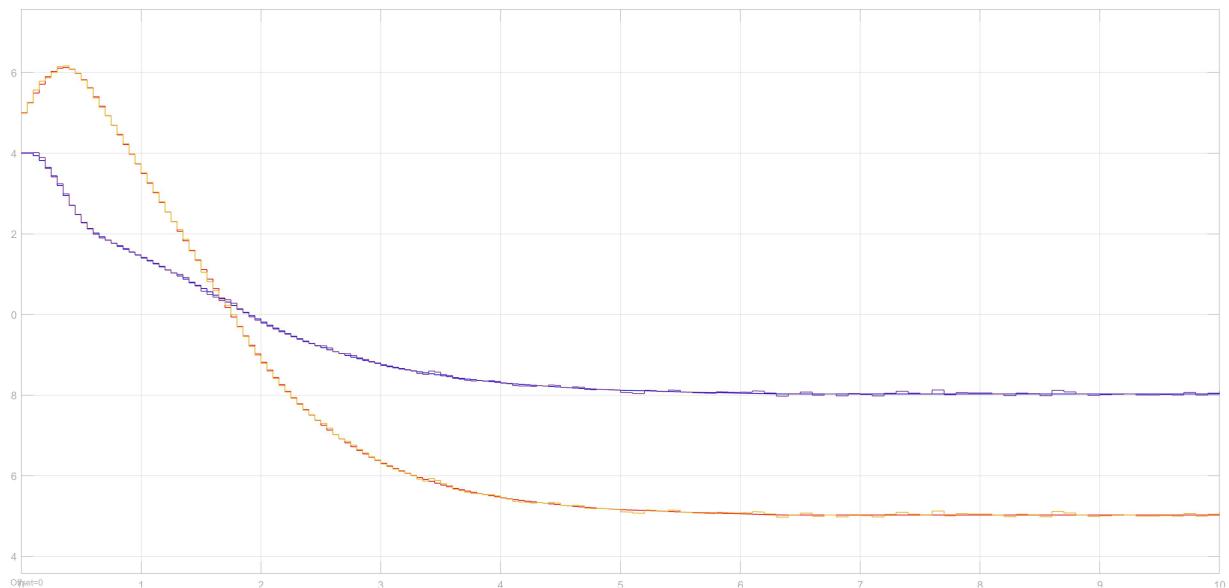


شکل ۲-۵: کنترل کننده تک ربات

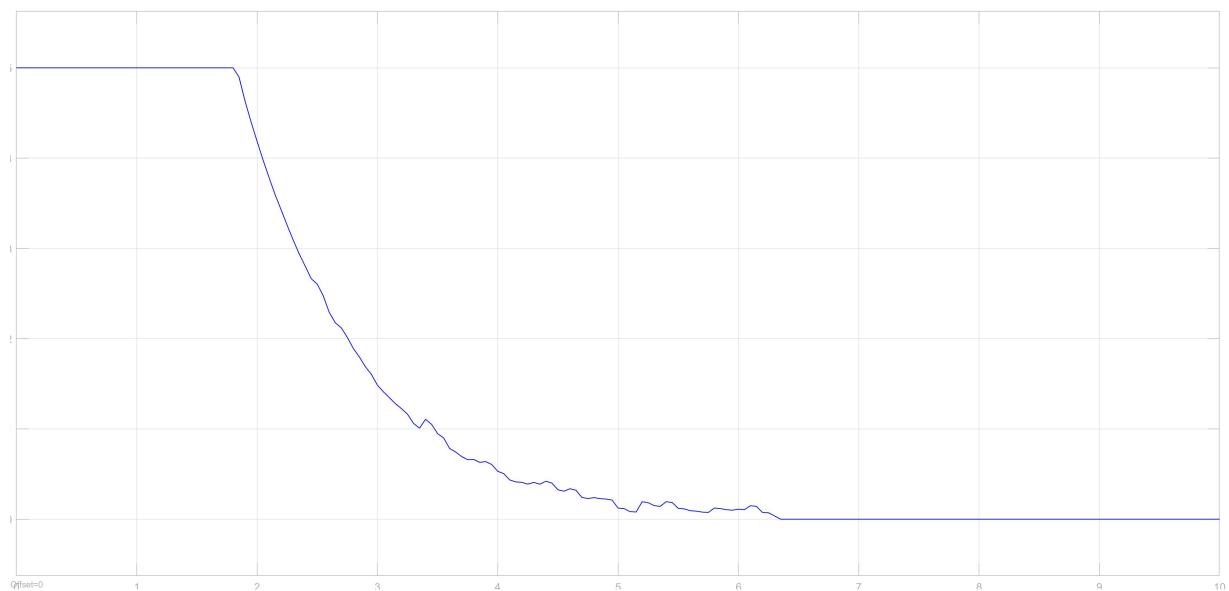


شکل ۲-۶: موقعیت مطلوب و موقعیت واقعی ربات

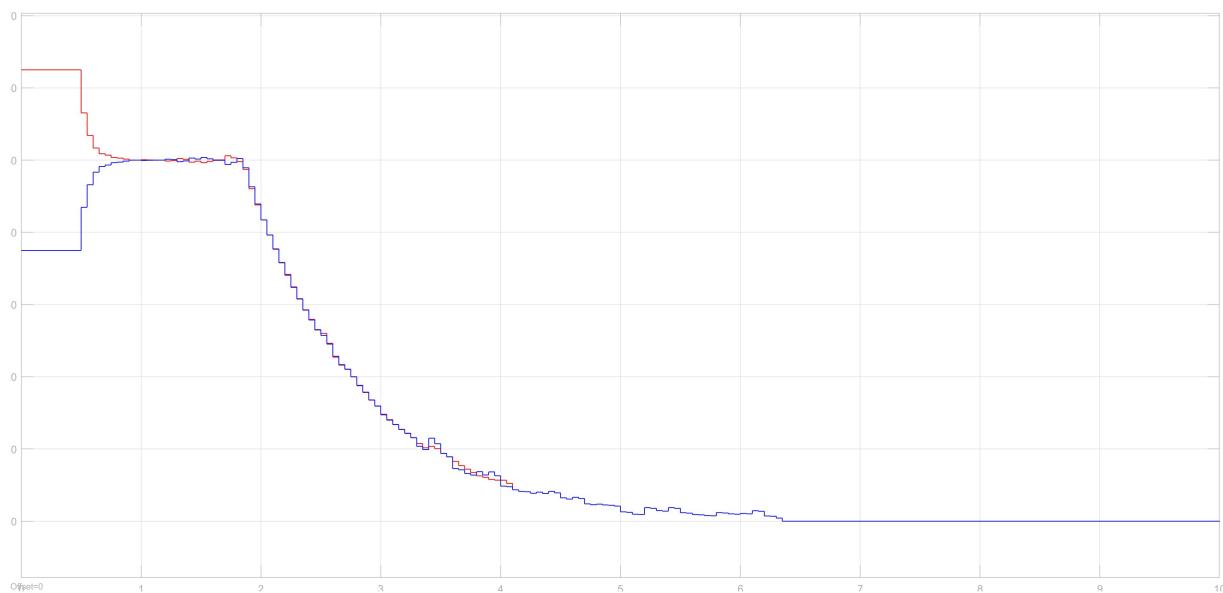
شکل های ۷-۲، ۸-۲ و ۹-۲ نیز نشان دهنده سرعت زاویه ای چرخ ها پس از رسیدن به هدف و حذف اثر نویز و محدود بودن سرعت خطی ربات هستند. همچنین شکل ۱۰-۲ نیز حرکت بدون مشکل ربات برای رسیدن به موقعیت مطلوب و در زاویه π - را نشان می دهد.



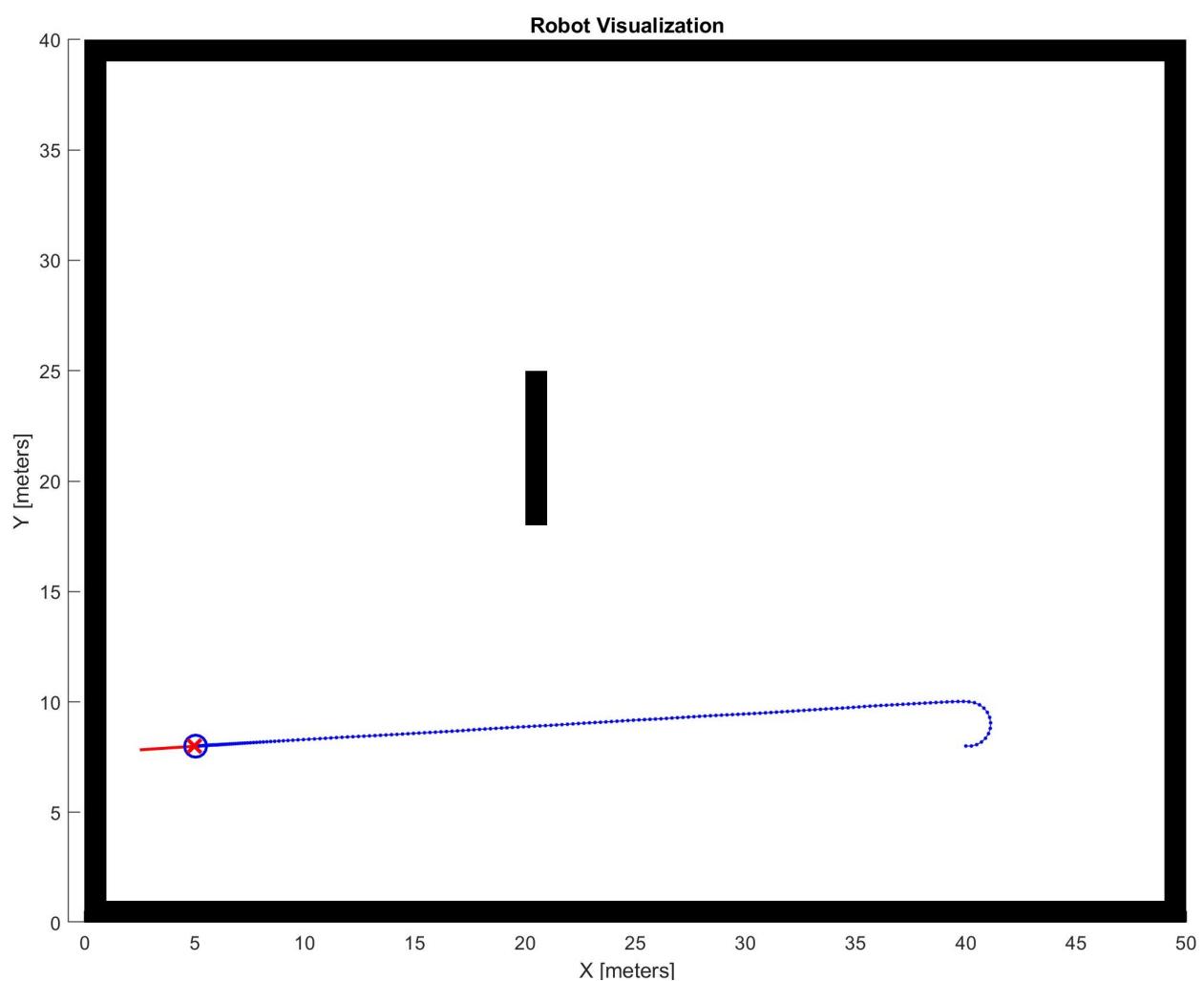
شکل ۷-۲: موقعیت واقعی و موقعیت فیدبک گرفته شده از سنسور ربات



شکل ۸-۲: سرعت خطی ربات



شکل ۹-۲: سرعت زاویه‌ای چرخ‌های ربات



شکل ۱۰-۲: مسیر پیموده شده توسط ربات در زاویه‌ای نزدیک به $-\pi$

فصل سوم

کنترل اجماع ربات‌ها

۱-۳ مقدمه

این فصل را می‌توان مهم‌ترین بخش این پژوهه دانست که هدف آن کنترل سیستم‌های چند عاملی^۱ به صورت دسته‌ای^۲ از ربات‌هاست، به طوری که رهبر^۳ دسته را دنبال کنند. در ادبیات سیستم‌های چند عاملی به هر خودرو یا ربات، عامل گفته می‌شود. همچنین همانطور که در ادامه خواهیم دید، یک خودرو فرضی نیز می‌تواند به عنوان رهبر در نظر گرفته شود. در سیستم‌های چند عاملی، هر عامل با داشتن اطلاعات عامل‌های دیگر، که می‌تواند بین یک تا تمام عامل‌های دیگر باشد، رهبر را دنبال می‌کند. در نتیجه، تمامی عامل‌ها درباره اینکه رهبر در کجا قرار دارد به اجماع^۴ برسند. به همین دلیل به الگوریتم‌های انجام دهنده این عمل، الگوریتم اجماع می‌گویند.

الگوریتم‌های اجماع تحت تاثیر ارتباطات بین عامل‌ها هستند و می‌توان با توجه به اینکه کدام ربات‌ها با هم داده تبادل می‌کنند، گراف ساده جهت‌داری^۵ رسم نمود. به این منظور هر عامل را به یک راس^۶ و هر ارتباط را یک یال^۷ جهت‌دار که از عامل فرستنده یال خارج و به عامل گیرنده وارد می‌شود، نسبت داد. همچنین، دور به طول یک، یالی که از یک راس به خودش وارد می‌شود، و تعداد بیشتر از یک یال^۸ بین دو راس مشخص وجود ندارد. هر گراف را می‌توان به نحوه‌های مختلفی، همچون ماتریس مجاورت^۹ و لیست مجاورت^۹ نمایش داد. در پژوهه با توجه به ادامه کار و نیازمنان، از ماتریس مجاورت استفاده شده است که دلیل آن را در ادامه خواهیم دید.

۲-۳ گراف اجماع

گراف اجماع تاثیر بسیار زیادی بر پژوهه و عملکرد آن دارد. هر یال (ارتباط)، به دلیل اضافه کردن محاسبات، شامل هزینه زمانی است و باعث بیشتر شدن زمان نمونه برداری می‌شود. از طرفی در صورت قطعی موقت ارتباط، نباید عامل ارتباط خود با دسته را به طور کامل از دست ندهد. در نتیجه مبحثی که پیش می‌آید این است که گرافی را طوری طراحی کنیم که در صورت قطع یک سری از ارتباطات

¹Multi-agent systems

²Platoon

³Leader

⁴Consensus

⁵Simple directed graph

⁶Vertex

⁷Edge

⁸Adjacency matrix

⁹Adjacency list

تضمين شود هیچ عاملی ارتباطش را با دسته از دست نمی‌دهد. در اين راستا از مباحث نظریه گراف بهره می‌بریم.

۱-۲-۳ عدد همبندی يالى

در ادبیات نظریه گراف مجموعه يالها و V مجموعه راس‌ها در نظر گرفته می‌شود [۱۷، ۳، ۵]. فرض کنید $G = (V, E)$ گرافی همبند باشد و $T \subseteq E$ باشد به طوری که $G - T$ ناهمبند باشد. در این صورت T را یک مجموعه برشی يالى (برش يالى)^{۱۰} گویند.

مجددا فرض کنید $G = (v, E)$ یک گراف ساده باشد، آنگاه G را k -همبند يالى^{۱۱} هرگاه $|V| > 1$ و برای هر $E \subseteq T$ که $|T| < k$ گرافی همبند باشد. از تعریف می‌توان نتیجه گرفت که اگر گرافی $k+1$ -همبند يالى باشد آنگاه k -همبند يالى است. در نتیجه بزرگترین مقدار k ، که گراف G به ازای آن k -همبند يالى باشد ارزش دارد و آن را (عدد) همبندی يالى^{۱۲} گراف G گویند و با $k'(G)$ نمایش می‌دهند.

برای به دست آوردن حد بالای عدد همبندی يالى، یکی از راس‌هایی که دارای کمترین درجه^{۱۳} در گراف است را در نظر بگیرید. درجه این راس را با $(G)\delta$ نمایش می‌دهند. اگر تمام يال‌های واردہ به این راس را حذف کنیم، به یک راس ایزوله می‌رسیم و چون در تعریف عدد همبندی يالى شرط وجود بیشتر از یک راس در گراف بود، گراف ناهمبند می‌شود زیرا مسیری بین این راس و راس‌های دیگر وجود ندارد. در نتیجه اثبات ذکر شده می‌توان نوشت:

$$k'(G) \leq \delta(G) \quad (1-3)$$

مزیت رابطه ۱-۳ در این است که به راحتی دید خوبی از بیشترین مقدار ممکن برای عدد همبندی يالى پیدا می‌کنیم.

اهمیت عدد همبندی يالى در گراف اجماع این است که اگر تعداد $1 - k'(G)$ ارتباط قطع شود، ارتباط غیر مستقیم دسته عامل‌ها همچنان حفظ می‌شود و هر یک می‌دانند که باید در چه موقعیتی قرار گیرند. پس هر چه قدر عدد همبندی يالى بزرگتر شود، ارتباط بین عامل‌ها مقاوم‌تر تلقی می‌شود.

¹⁰Edge cut (set)

¹¹ k -edge connected

¹²Edge connectivity

¹³degree

۲-۲-۳ کمینه‌ترین گراف اجماع

همانطور که صحبت شد، بیشتر شدن یال‌ها موجب افزایش زمان نمونه برداری می‌شود. از طرفی بزرگ بودن عدد همبندی یالی نیز به عنوان معیاری از مقاوم بودن گراف در برابر قطع شدن ارتباط‌ها می‌باشد. طبق تعاریف انجام شده در قسمت ۱-۲-۳^{۱۴}، می‌توان گفت که اضافه کردن یال عدد همبندی یالی را کاهش نمی‌دهد و در بعضی از شرایط موجب افزایش آن می‌شود. پس در تشکیل گراف اجماع، نکته قابل توجه، انتخاب برقرار کردن یا نکردن ارتباط بین هر دو عامل است به طوری که کمینه‌ترین ارتباط با توجه به عدد همبندی یالی مورد نظرمان برقرار شود.

در حالت ایده‌آل، اگر اطمینان داده شود که هیچ ارتباطی قرار نیست از دست رود، می‌توان عدد همبندی یالی را برابر یک در نظر گرفت. یک درخت^{۱۵} با ریشه^{۱۶} رهبر، کمینه‌ترین ارتباط را به ازای $k' = 1$ برقرار می‌کند. منظور از کمینه‌ترین ارتباط در این حالت، این است که اولاً گراف ضعیفاً همبند^{۱۷} باشد، به این معنا که با جایگزینی یال‌های جهت‌دار با یال‌بی‌جهت بین هر دو عامل (راس)، مسیری وجود داشته باشد. از طرفی منظور از کمینه بودن، این است که با حذف هر ارتباط، همبندی و در نتیجه ارتباط غیر مستقیم با رهبر از دست بروود. در صورت وجود دور در گراف، می‌توان یکی از یال‌ها را حذف نمود و هنوز ارتباط بین عامل‌ها حفظ شود. در نتیجه گراف ما نباید دور داشته باشد. به گراف بدون دور همبند، درخت گفته می‌شود. همچنین به گرافی کمینه‌تر از درخت نمی‌شود رسید زیرا در صورت قطع ارتباط، بین هر دو عامل، عامل فرزند^{۱۸}، راسی که ارتباط به آن وارد شده، و فرزندانش ارتباط خود را با والد^{۱۹}، راسی که ارتباط از آن خارج شده، و در نهایت رهبر از دست می‌دهند.

در حالتی که بخواهیم شبکه گراف، بیشترین مقاومت را نسبت به قطعی ارتباطات داشته باشد، یا به عبارتی به بیشترین مقدار ممکن برای عدد همبندی یالی برسیم، به طوری که گراف کمینه هم باشد، باید شبکه ارتباطات را به صورت گراف کامل ایجاد کرد. در ادامه ادعای بالا اثبات می‌شود.

تعداد عامل‌ها را برابر با n می‌گیریم. پس داریم:

$$|V| = n \quad (2-3)$$

¹⁴Tree

¹⁵Root

¹⁶Weakly connected

¹⁷Child

¹⁸Parent

با توجه به اینکه گراف ساده است پس حداکثر، از هر راس به تمامی رئوس دیگر می‌توان یال داشت. در نتیجه اگر همه رئوس به هم متصل باشند یا به عبارتی، درجه همه رئوس برابر با $n - 1$ باشد، آنگاه گراف کامل است و گراف کامل n راسی را با K_n نشان می‌دهیم. در نتیجه داریم:

$$\delta(K_n) = n - 1 \quad (3-3)$$

از روابط ۱-۳ و ۳-۳ می‌توان نتیجه گرفت که:

$$k'(K_n) \leq n - 1 \quad (4-3)$$

پس اگر گراف کامل نباشد، یعنی یک راسی وجود دارد که از آن به همه رئوس دیگر، یال وجود ندارد پس درجه آن راس کمتر از $n - 1$ و در نتیجه $n - 1 < \delta(G)$ خواهد بود. مجدداً با توجه به رابطه ۱-۳، می‌توان گفت که $n - 1 < k'(G)$ خواهد بود. حال اگر اثبات شود که

$$k'(K_n) = n - 1 \quad (5-3)$$

می‌توان نتیجه گرفت که به ازای یک $n - 1$ ثابت به عنوان تعداد رئوس یک گراف ساده، به بیشترین مقدار ممکن برای عدد همبندی یالی می‌رسیم، اگر و فقط اگر آن گراف، کامل باشد. برای اثبات رابطه ۵-۳، از برهان خلف استفاده می‌کنیم. فرض کنید رابطه برقرار نباشد پس با توجه به رابطه ۴-۳، $k'(K_n) < n - 1$ است، زیرا از بالا محدود شده است. پس طبق تعریف، دو راس $\exists u, v \in V(K_n)$ وجود دارند، به طوری که با حذف کمتر از $n - 1$ یال، گراف ناهمبند می‌شود. حال مسیر uv و تمام مسیرهای uvw را در نظر بگیرید به این صورت که w رئوس دیگر گراف K_n باشد. برای w ، $2 - n$ انتخاب داریم. پس تعداد کل مسیرهای ذکر شده برابر $n - 1$ است و همگی از هم مستقل یالی می‌باشند. به این معنی که هیچ یالی در این مسیرها تکرار نشده است. در نتیجه وجود مجموعه یالهایی که تعداد آن‌ها کمتر از $n - 1$ باشد و باعث شوند همگی این مسیرها از بین روند تا درنهایت بین دو راس u و v هیچ مسیری نباشد، تناقض است. پس فرض خلف باطل و حکم ثابت می‌شود. در نتیجه، قوی‌ترین گراف برای تشکیل ارتباطات و همچنین کمینه‌ترین آن، گراف کامل می‌باشد. حال به تشکیل گراف اجماع می‌پردازیم.

۳-۲-۳ تشکیل گراف اجماع

در این پژوهه آزمایشات بر روی ۴ ربات انجام شده است که رهبر مجازی است و تنها ۳ ربات وجود فیزیکی دارند. با توجه به بخش ۲-۲-۳ می‌توان گفت در صورتی که تنها ارتباط دو طرفه بین عامل رهبر و بقیه عامل‌ها برقرار شود به کمینه‌ترین گراف اجماع خواهیم رسید که دارای ۳ یال خواهد بود. با وجود نکته بالا، به دلیل اینکه گراف مقاوم‌تری داشته باشیم، از گراف کامل K_4 استفاده شده است.

۳-۳ الگوریتم اجماع

در این پایان‌نامه از دینامیک تک انتگرال‌گیر^{۱۹} برای ارتباط بین موقعیت‌های ربات‌ها استفاده شده است. می‌توان این الگوریتم را به صورت زیر در نظر گرفت:

$$\dot{x}_i = - \sum_{j=1}^n a_{ij}(x_i - x_j) \quad (6-3)$$

که در آن a_{ij} اگر از راس زبه i یالی وارد شده باشد. در غیر این صورت $a_{ij} = 0$ خواهد بود. به عنوان مثال اگر داشته باشیم:

$$\dot{x}_3 = -(x_3 - x_1) - (x_3 - x_2) \quad (7-3)$$

به این معنی است که داده عامل ۱ و ۲ به ۳ می‌رسد. اگر x را موقعیت تعریف کنیم، اثبات می‌شود که تحت ماتریس‌بندی خاصی با گذشت زمان، موقعیت عامل‌ها به هم خواهند رسید. در نتیجه اگر یکی از ربات‌ها رهبر باشد، بقیه به موقعیت آن خواهند رسید.

علاوه بر دینامیک تک انتگرال‌گیر، مدل‌های دو انتگرال‌گیر^{۲۰}، مدل خطی یا غیر خطی نیز می‌توانند استفاده گردند. برای سادگی مدل تک انتگرال‌گیر در این پایان‌نامه استفاده شده است.

اگر ماتریسی که با a_{ij} ساخته می‌شود را A نامیم، ماتریس لaplacian^{۲۱} آن به صورت زیر تعریف

¹⁹Single integrator

²⁰Double integrator

²¹Laplacian matrix

می‌شود:

$$L(A) = D(A) - A \quad (8-3)$$

که در آن $(A)D$ ماتریس درجه گراف A است که در آن مجموع درجات هر راس بر روی قطر اصلی، متناظر با سطر آن راس، نوشته می‌شود. برای سادگی ماتریس‌های $L(A)$ و $D(A)$ ، با L و D نوشته خواهند شد. می‌توان رابطه ۸-۳ را به فرم ماتریسی، به صورت زیر نوشت:

$$\dot{x} = -Lx \quad (9-3)$$

که نتیجه می‌دهد:

$$x(t) = e^{-Lt}x(0) \quad (10-3)$$

در این پروژه، با توجه به استفاده از ماتریس کامل به عنوان گراف اجماع برای ارتباط ۴ ربات داریم:

$$A = K_4 = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \quad (11-3)$$

که در نتیجه آن ماتریس لاپلاسین با توجه به رابطه ۸-۳ به صورت زیر محاسبه می‌گردد:

$$L = D - A = \begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 3 \end{bmatrix} - \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 3 & -1 & -1 & -1 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 3 & -1 \\ -1 & -1 & -1 & 3 \end{bmatrix} \quad (12-3)$$

در ادامه به خواص ماتریس لاپلاسین می‌پردازیم.

۱-۳-۳ خواص ماتریس لاپلاسین

ماتریس لاپلاسین خواص مهمی مخصوصا در مقادیر ویژه خود دارد که در ادامه تعدادی از آن‌ها ذکر می‌گردند [۱۴]:

۱. ماتریس L یک ماتریس متقارن است. با توجه به اینکه A و D متقارن‌اند پس نتیجه می‌شود حاصل تفاضل آن‌ها، نیز متقارن خواهد بود.

$$L = L^T$$

۲. جمع هر سطر برابر با صفر است. با توجه به اینکه در سطر i ام داریم a_{ij} و $d_i = \sum_{j=1}^n a_{ij}$ و سپس ماتریس‌های A و D از هم کم می‌شوند، نتیجه می‌دهد که مجموع هر سطر برابر صفر خواهد شد.

۳. ماتریس L یک ماتریس مثبت شبه معین^{۲۲} است. ماتریس مربعی B را یک ماتریس مثبت شبه معین گویند اگر برای هر $x \in \mathbb{R}^n$ داشته باشیم:

$$x^T B x \geq 0 \quad (13-3)$$

همچنین ماتریس مربعی B را مثبت معین^{۲۳} گویند اگر و فقط اگر برای هر $x \in \mathbb{R}^n$ داشته باشیم:

$$x^T B x > 0 \quad (14-3)$$

اثبات این خاصیت با بسط دادن عبارت $x^T B x$ به دست می‌آید.

۴. با توجه به اینکه L یک ماتریس مثبت شبه معین است، در نتیجه مقادیر ویژه آن نامنفی و حقیقی هستند.

۵. اگر گراف G ، k مولفه همبندی داشته باشد، آنگاه دقیقا k تا از مقادیر ویژه آن برابر صفر خواهند بود. به ازای گراف همبند که فقط یک مولفه همبندی دارد اگر برداری که همه مولفه‌های آن باشند را در آن ضرب کنیم، هر المان برابر با مجموع هر راس می‌شود، در نتیجه به بردار صفر

²²positive semi-definite

²³positive definite

می‌رسیم.

$$L \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^n l_{1j} \\ \sum_{j=1}^n l_{2j} \\ \vdots \\ \sum_{j=1}^n l_{nj} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (15-3)$$

برای اثبات k مولفه همبندی نیز، اگر راس‌های هر مولفه همبندی را کنار هم قرار دهیم، به تعدادی بلوک مستقل از هم می‌رسیم که هر کدام از یکدیگر مستقل‌اند و در هر کدام می‌توان یک مقدار ویژه صفر، با ایجاد برداری که به ازای سطرهای آن بلوک مقدار ۱ و بقیه سطرهای مقدار صفر داشته باشد، همانند شکل ۱-۳، رسید.

$$\left(\begin{array}{ccccc} L_1 & & & & \\ & 0 & & & \\ & & L_2 & & \\ & & & \ddots & \\ & & & & L_k \end{array} \right) \begin{bmatrix} 1 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

شکل ۱-۳: بردار ویژه متناظر با مقدار ویژه صفر در ماتریس لاپلاسین گراف دارای k مولفه همبند

۶. گراف همبند که دارای یک مولفه همبندی است، تنها یک مقدار ویژه برابر با صفر دارد که بردار متناظر با آن برداری است که تمام المان‌های آن ۱ باشد.

حال که خواص ماتریس لاپلاسین بررسی شد، در قسمت بعد پایداری رابطه ۹-۳ اثبات می‌گردد.

۲-۳-۳ پایداری کنترل اجماع

در کنترل اجماع، موقعیت عامل رهبر، x_m به سیستم وارد می‌شود و کنترل اجماع تاثیری بر روی آن ندارد و \dot{x}_i از طریق الگوریتم مسیریابی تعیین می‌گردد. لذا در رابطه ۹-۳ سطر اول L برابر صفر می‌گردد. به عنوان مثال در این پژوهه، L جدید، طبق رابطه ۱۲-۳ برابر است با:

$$L = \begin{bmatrix} 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 3 & -1 \\ -1 & -1 & -1 & 3 \end{bmatrix} \quad (16-3)$$

اگر x برای مسئله یک رهبر و n عامل دیگر به صورت زیر تعریف گردد:

$$x = \begin{bmatrix} x_m \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad (17-3)$$

می‌توان x_m را به صورت ورودی سیستم در نظر گرفت و سیستم را به شکل زیر بازنویسی کرد:

$$\dot{x} = Ax + Bu, u = x_m \quad (18-3)$$

که در آن A و B به شکل زیر از روی درایه‌های ماتریس L به دست می‌آیند:

$$A = \begin{bmatrix} l_{22} & l_{23} & \cdots & l_{2n} \\ l_{32} & l_{33} & \cdots & l_{3n} \\ \vdots & & & \\ l_{n2} & l_{n3} & \cdots & l_{nn} \end{bmatrix}, B = \begin{bmatrix} l_{21} \\ l_{31} \\ \vdots \\ l_{n1} \end{bmatrix} \quad (19-3)$$

چون ماتریس اولیه لاپلاسین بوده پس اگر مقادیر هر سطر B را از درایه قطر اصلی ماتریس A که

همان درایه قطر اصلی ماتریس لاپلاسین اولیه است کم کنیم به یک ماتریس لاپلاسین می‌رسیم. از اینجا به بعد منظور از ماتریس L ماتریسی است که در تعریف زیر آمده است.

$$A + diag(B) = -L, \quad diag(B) = \begin{bmatrix} B_1 & 0 & \cdots & 0 \\ 0 & B_2 & \cdots & 0 \\ \vdots & & & \\ 0 & 0 & \cdots & B_n \end{bmatrix} \quad (20-3)$$

همچنین در ادامه منظور از ماتریس $\mathbb{1}$ ماتریسی است که تمام درایه‌های آن ۱ باشند.
ماتریس خط، e ، به صورت زیر تعریف می‌شود:

$$e = x - \mathbb{1}x_m \quad (21-3)$$

به کمک روابط ۱۸-۳ و ۲۰-۳ داریم:

$$\dot{x} = Ax + Bx_m = -Lx - diag(B) + Bx_m \quad (22-3)$$

با ادغام روابط ۲۱-۳ و ۲۲-۳ می‌توان نوشت:

$$\dot{x} = -Le + L\mathbb{1}x_m - diag(B)e \quad (23-3)$$

با توجه به رابطه ۱۵-۳، رابطه بالا به صورت زیر تبدیل می‌گردد:

$$\dot{x} = -Le - diag(B)e \quad (24-3)$$

که با کمک از رابطه ۲۰-۳ می‌توان به رابطه زیر رسید:

$$\dot{x} = Ae \quad (25-3)$$

در نهایت چون بر متغیر \dot{x}_m کنترلی نداریم و قرار است مطابق الگوریتم مسیریابی مقدار گیرد، آن را

در هر فریم موقعیت عامل رهبر را ثابت می‌گیریم و در نتیجه می‌توان از مشتق آن صرف نظر کرد. با این فرض، سیستم دارای اندکی تاخیر می‌شود ولی در ادامه با اثبات پایدار مجانبی فراگیر^{۲۴} بودن سیستم، مطمئن خواهیم بود که هنگام توقف خطا صفر خواهد بود. در نتیجه فرض بالا داریم:

$$\dot{x} = \dot{e} \quad (26-3)$$

با استفاده از دو رابطه ۲۶-۳ و ۲۴-۳ نتیجه می‌شود:

$$\dot{e} = -Le - \text{diag}(B)e \quad (27-3)$$

حال تابع کاندید لیاپانوف^{۲۵}, $V(e)$, به صورت زیر تعریف می‌گردد،

$$V(e) = \frac{1}{2}e^T e > 0 \quad (28-3)$$

مشتق آن برابر است با:

$$\dot{V}(e) = e^T \dot{e} \quad (29-3)$$

با جایگذاری رابطه ۲۷-۳ در ۲۹-۳ خواهیم داشت:

$$\dot{V}(e) = -e^T Le - e^T \text{diag}(B)e \quad (30-3)$$

حال به بررسی رابطه ۳۰-۳ می‌پردازیم. اولاً طبق خواص مطرح شده در قسمت ۱-۳-۳ و ماتریس مثبت معین بودن هر ماتریس لاپلاسینی داریم: $\forall x, L x^T Lx \geq 0$ در نتیجه در رابطه ۳۰-۳ داریم:

$$-e^T Le \leq 0 \quad (31-3)$$

²⁴Globally asymptotically stable

²⁵Lyapunov candidate function

از طرفی با توجه به تعریف درایه‌های B در رابطه ۱۹-۳ که هیچ یک درایه‌های قطر اصلی ماتریس لaplacian نیستند، درایه‌های B می‌توانند یا صفر و یا یک باشند ولی حداقل یک درایه یک داریم، زیرا گراف همبند است و باید از عامل رهبر ارتباطی با بقیه موجود باشد. لذا اگر اندیس عامل‌هایی را که با عامل رهبر در ارتباط اند در مجموعه N نگهداری کنیم داریم:

$$-e^T \text{diag}(B)e = -\sum_{j \in N} e_j^2 \leq 0 \quad (32-3)$$

در نتیجه با توجه به روابط ۳۰-۳، ۳۱-۳ و ۳۲-۳ داریم:

$$\dot{V}(e) \leq 0 \quad (33-3)$$

حال اثبات می‌کنیم که سیستم بالا پایدار مجانبی فراگیر است.

قضیه ۳-۳-۱. سیستم پایدار مجانبی است، اگر تنها یک نقطه تعادل پایدار داشته و تابع اسکالر $V(e)$ وجود داشته باشد که شرایط زیر را برآورده سازد [۱۲]:

۱. $V(e)$ در تمامی فضای حالت پیوسته و مشتق‌های جزئی پیوسته دارد.

$$x \neq 0 \quad V(e) > 0 \quad .2$$

$$V(0) = 0 \quad .3$$

$$\|x\| \rightarrow \infty \quad V(e) \rightarrow \infty \quad .4$$

$$.\quad .5 \quad \dot{V}(e) \leq 0 \quad \text{و پاسخ غیر صفر} \quad \dot{V}(e) = 0 \quad \text{پاسخی از معادله دیفرانسیل سیستم نیست.}$$

شرط ۱ تا ۴ توسط رابطه ۲۸-۳ به دست می‌آیند. قسمت اول شرط ۵، در رابطه ۳۳-۳ به دست آمده است. در ادامه قضیه ۲-۳-۳ را اثبات می‌کنیم.

قضیه ۳-۳-۲. اگر $e \neq 0$ وجود داشته باشد به طوری که $\dot{V} = 0$ باشد، آنگاه پاسخی از معادله دیفرانسیل سیستم نیست.

با برهان خلف قضیه را اثبات می‌کنیم. فرض خلف: فرض می‌کنیم e_j وجود دارد به ظوری که $\dot{V}(e) = 0$. تا به الان از فرض مهمی که در اثبات پایداری استفاده نشده است، همبند بودن گراف است.

گفته شد چون گراف باید همبند باشد، پس از عامل رهبر به حداقل یکی از عامل‌ها باید ارتباط مستقیم وجود داشته باشد. در نتیجه حداقل یکی از درایه‌های بردار B برابر ۱ است. یکی از آن درایه‌ها را انتخاب می‌کنیم $0 = B_i$. با توجه به رابطه ۳۰-۳ اگر $e_i \neq 0$ باشد، آنگاه عبارت $e^T B e < 0$ خواهد بود که به همراه ۳۱-۳ نتیجه می‌دهد $\dot{V}(e) < 0$ و به تناقض می‌رسیم. در نتیجه $0 = e_n$ باید باشد. حال چون $0 \neq e$ در نتیجه درایه‌ای همانند $0 \neq e_j$ وجود دارد. حال از روی اینکه $0 = \dot{e}$ یا در حالت کلی

$$\forall i = 1, 2, \dots, n \quad \dot{e}_i = 0 \quad (34-3)$$

باشد به تناقض خواهیم رسید. چون گراف همبند است پس هیچ راسی دارای درجه صفر نیست، لذا تمام درایه‌های قطر اصلی ماتریس A غیر صفر هستند. اگر فقط درایه $0 \neq e_j$ داشته باشیم، در نتیجه چون درایه قطر اصلی متناظر با سطر j مخالف صفر است، $0 \neq \dot{e}_j$ می‌شود. پس اگر W مجموعه‌ی اندیس تمامی عامل‌های غیر لیدر باشد آنگاه J را کوچک‌ترین مجموعه‌ای در نظر می‌گیریم به طوری که از اندیس‌ها باشد به طوری که

$$\forall j \in J \quad e_j \neq 0 \quad (35-3)$$

$$\dot{e} = 0.$$

دلیل اینکه امکان ندارد $J = W$ باشد این است که باید $0 = e_y$ باشد.

حال دو حالت داریم: یا تمامی e_{ji} ‌ها با هم برابرند یا حداقل یک جفت وجود دارند که با هم ارتباط مستقیم دارند و در آن یکی از دیگری بزرگتر است.

اگر تمامی e_{ji} با هم برابر و دارای مقدار c باشند، به معنی آن است که اگر یک بردار z تشکیل دهیم که متناظر با سطر ji مقدار c و در بقیه سطرها مقدار صفر بگیرد $z = e$ ، طبق رابطه ۲۷-۳ داریم:

$$\dot{e} = -Lz - diag(B)z = 0 \quad (36-3)$$

و چون سطرهایی که $z_i = 0$ است، خواهیم داشت:

$$diag(B)z = 0 \quad (37-3)$$

با ادغام روابط ۳۶-۳ و ۳۷-۳ داریم:

$$Lz = 0 \quad (38-3)$$

که نتیجه می‌دهد z بردار ویژه L است و چون $\mathbb{1} \neq z$ پس L حداقل ۲ مقدار ویژه مخالف صفر دارد که این با همبند بودن گراف اجماع در تناقض است. در نتیجه برای اینکه حالت بالا پیش نیاید جفت $|e_{j2}| > |e_{j1}|$ در مجموعه J وجود دارند که دارای ارتباط مستقیم هستند. نکته قابل توجه این است که در سطرهای عناصر J ، درایه قطر اصلی برابر منفی مجموع درایه‌های دیگر است. در نتیجه اگر قدرمطلق درایه قطر اصلی در سطر $J \in i$ برابر k_i باشد داریم:

$$k_i e_i = \sum_{t \in T_i} e_t, \quad |T_i| = k_i \quad (39-3)$$

که در آن T_i مجموعه هایی در سطر i است که $a_{it} = 1$ باشد. حال اگر سطر j را در نظر بگیریم، با توجه به رابطه ۳۹-۳ و $j \in T_{j2}$ داریم:

$$\exists j \in T_{j2} \quad |e_{j3}| > |e_{j2}| \quad (40-3)$$

رابطه ۴۰-۳، مربنا تکرار می‌شود و به e_{j4} و بقیه e_{ji} ها می‌رسیم. در نهایت چون یک رابطه خوش‌ترتیبی بیان گشته و مجموعه W و در نتیجه آن J متناهی است، عضوی همانند e_{jx} وجود دارد که عضوی بزرگتر از اندازه آن در J و چون بقیه e_i ها نیز برابر صفر هستند، عضوی بزرگتر از اندازه آن در بین کل e_i ها وجود ندارد. همچنین چون

$$\exists j(x-1) \in T_{jx} \quad s.t. \quad |e_{j(x-1)}| < |e_{j(x)}| \quad (41-3)$$

طبق رابطه ۳۹-۳ در سطر jx داریم:

$$k_{jx} e_{jx} = e_{j(x-1)} + \sum_{t \neq j(x-1) \in T_{jx}} e_t, \quad |T_{jx}| = k_{jx} \quad (42-3)$$

اگر از دو طرف رابطه $42-3$ ، قدر مطلق بگیریم داریم:

$$k_{jx}|e_{jx}| = |e_{j(x-1)}| + \sum_{t \neq j(x-1) \in T_{jx}} |e_t|, \quad |T_{jx}| = k_{jx} \quad (43-3)$$

که با در نظر گرفتن روابط $41-3$ و $43-3$ ، نتیجه می‌شود که

$$\exists t \in T_{jx} \text{ s.t. } |e_t| > |e_{jx}| \quad (44-3)$$

و این عبارت تناقض است.

در نتیجه تمامی شرایطی که در نظر گرفتیم، به تناقض رسید، پس فرض خلف باطل و حکم که قضیه $2-3-3$ است، ثابت می‌گردد.

در نتیجه تمامی شرایط قضیه $1-3-3$ برقرار می‌گیرد و سیستم پایدار مجانبی فرآگیر خواهد بود و خطای صفر خواهد رسید.

۳-۳-۳ روش شکل‌دهی

با رابطه $9-3$ به این رسیدیم که اگر یکی از عامل‌ها را رهبر در نظر بگیریم، بقیه آن را دنبال خواهند کرد. حال با یک حقه، ماتریس شکل‌دهی^{۲۶} را تعریف و اعمال می‌کنیم. نسبت اختلاف موقعیت هر ربات را در F که ماتریس شکل‌دهی است، قرار می‌دهیم و با رابطه $45-3$ ، کاری می‌کنیم که موقعیت واقعی عامل i با اختلاف F_i دیده شود و در نتیجه وقتی عامل میخواهد به موقعیت مطلوب x_{di} برود در واقعیت به موقعیت $x_{di} + F_i$ خواهد رسید که همان هدف ما از شکل‌دهی است. این روش با اندکی تغییر نسبت به [۶] پیاده‌سازی شده است.

$$\dot{x} = -L(x - F) \quad (45-3)$$

همچنین چون در دو بعد $y-x$ مشغول به کار هستیم، بردار x را به ماتریس با دو ستون تغییر دادیم به طوری که یک ستون مربوط به فاصله در محور x و یک ستون مربوط به فاصله در محور y باشد. و در

²⁶formation

نتیجه ماتریس شکل‌دهی نیز دارای ۲ ستون است.

$$X = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots \\ x_n & y_n \end{bmatrix} \quad (46-3)$$

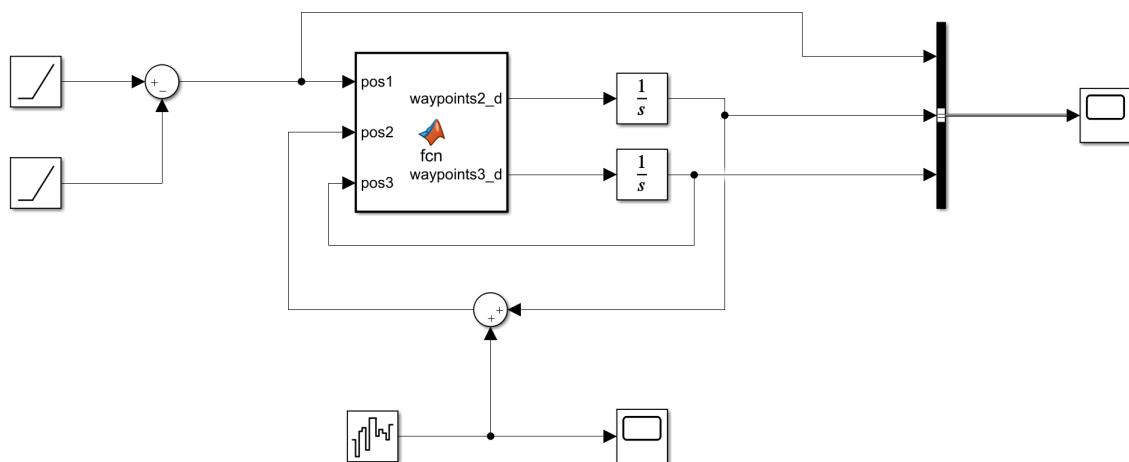
تنها نکته مهم این است که برای کاهش تاخیر ایجاد شده در کنترل کننده‌ها و خود کنترل کننده پلتون، از رهبر مجازی استفاده شده است.

۴-۳ شبیه‌سازی و نتایج

در این قسمت، ابتدا شبیه‌سازی اجماع به صورت عددی و سپس همراه با ربات‌ها انجام خواهد شد.

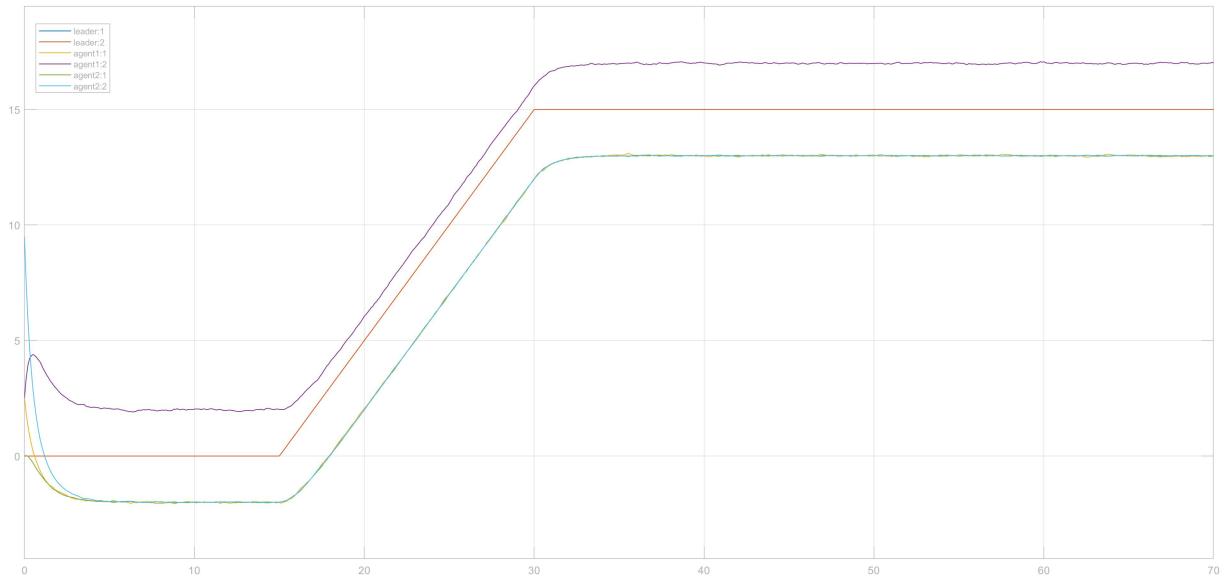
۱-۴-۳ شبیه‌سازی اجماع عددی

بلوک‌های شبیه‌سازی شده به شکل ۲-۳ هستند.



شکل ۲-۳: بلوک‌های شبیه‌سازی شده برای اجماع عددی

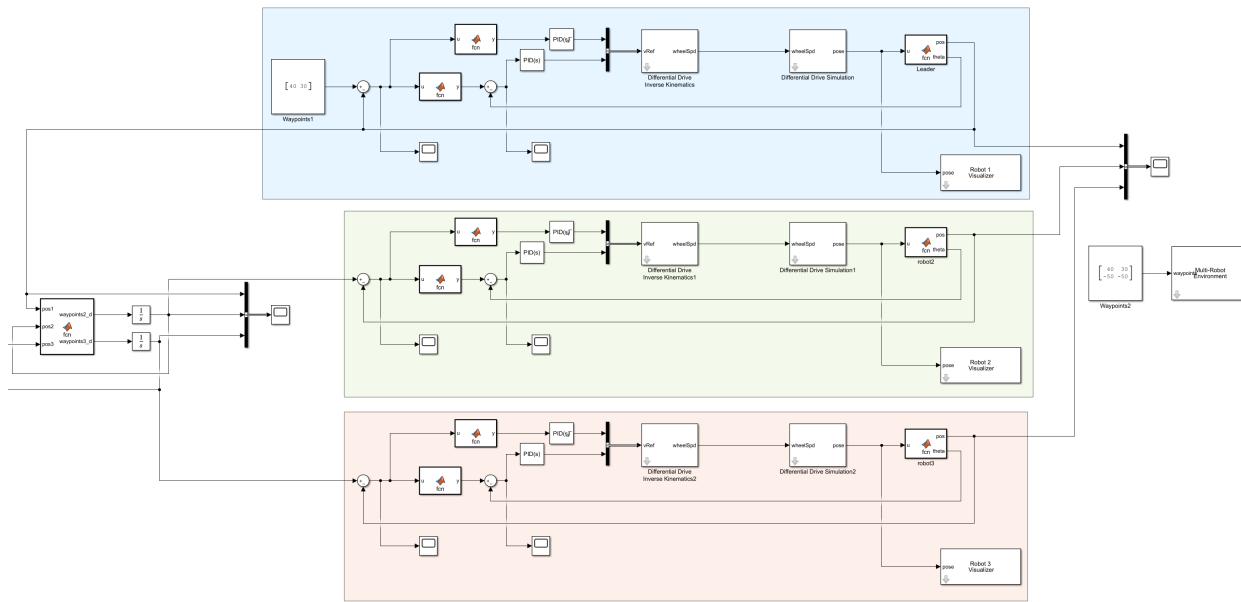
همچنین موقعیت عامل‌ها با اعمال نویز به همراه عامل رهبر در حال جابه‌جایی، در شکل ۳-۳ آمده است.



شکل ۳-۳: موقعیت مکان‌های شبیه‌سازی شده

۲-۴-۳ شبیه‌سازی با ربات‌ها

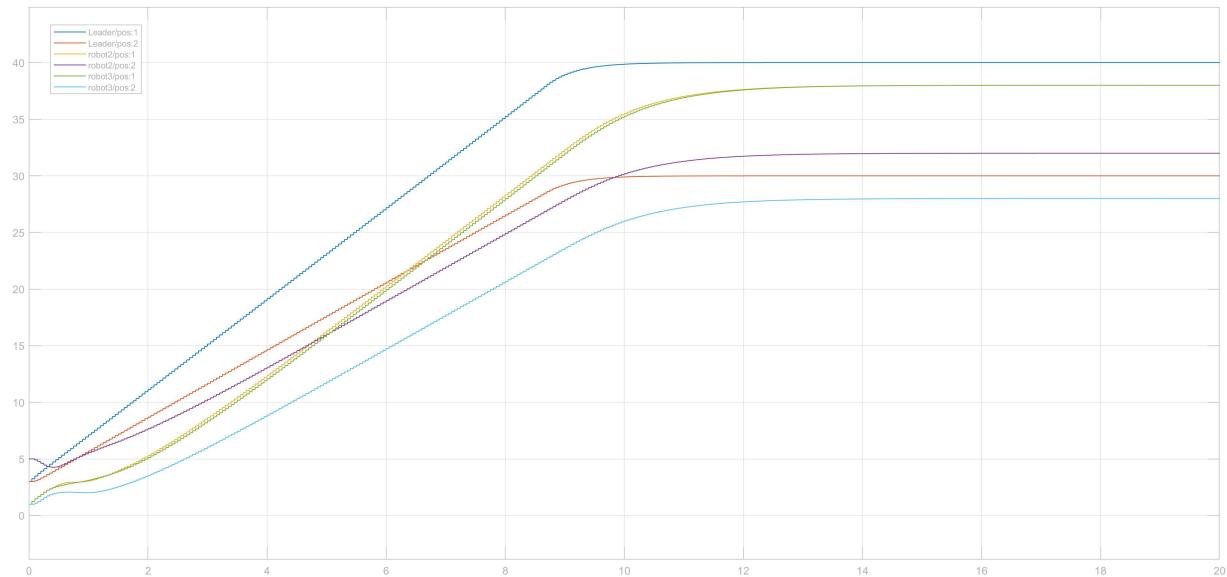
در این قسمت دو شبیه‌سازی، اول با رهبر واقعی که یکی از ربات‌ها است و در شبیه‌سازی دوم با رهبر مجازی، برای کم کردن تاخیر، خواهیم دید. شبیه‌سازی اجماع با رهبر واقعی در شکل ۴-۳ آمده است.



شکل ۴-۳: بلوک‌های شبیه‌سازی شده برای اجماع ربات‌ها با رهبر واقعی

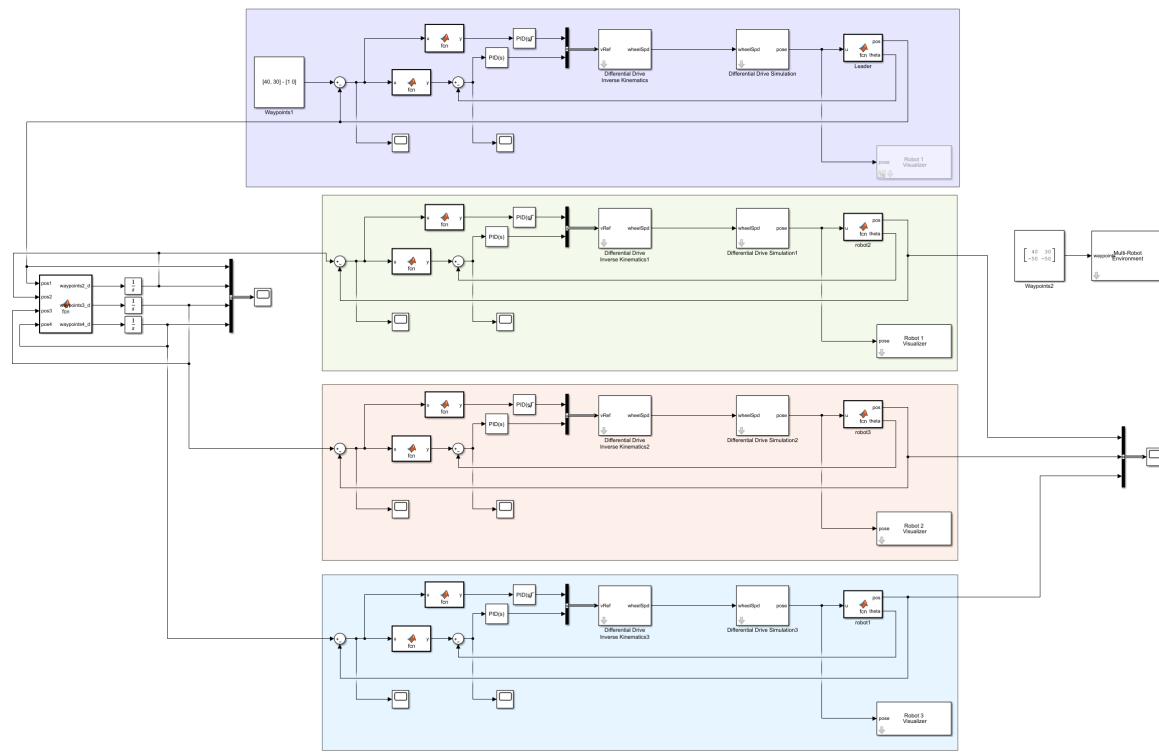
در شکل ۵-۳ مکان ربات‌ها در صفحه آمده است. همانطور که مشاهده می‌شود، ربات رهبر در حدود

ثانیه ۱۰ به موقعیت مطلوب رسیده است، در حالی که بقیه ربات‌ها در حدود ۱۲ ثانیه به نقطه هدف خود رسیده‌اند.

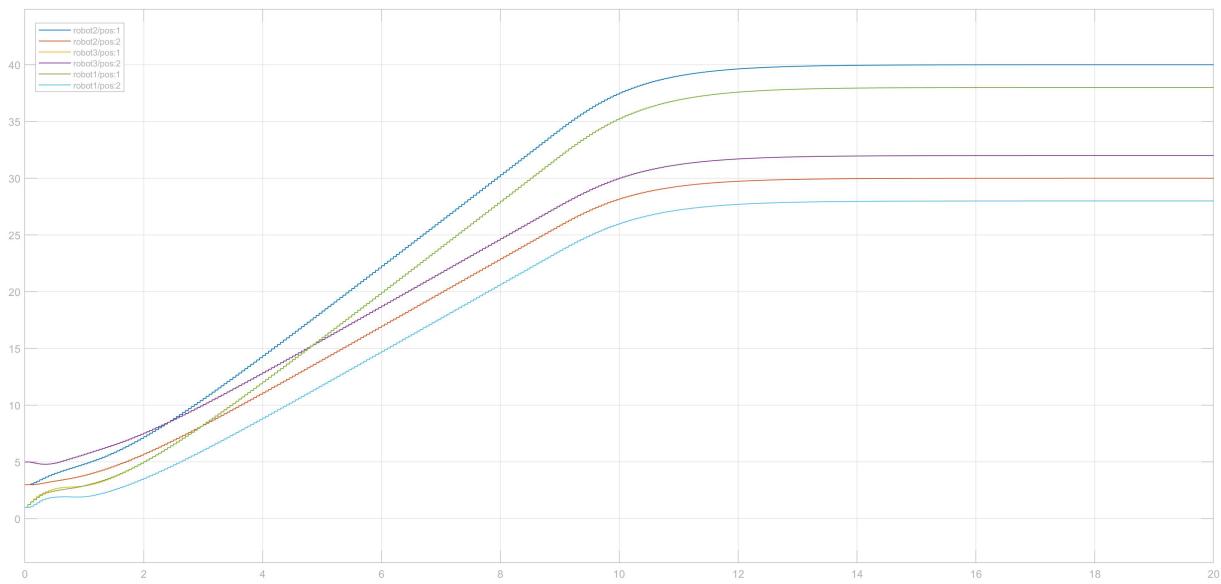


شکل ۳-۵: موقعیت مکان‌های ربات‌ها برای اجماع با رهبر واقعی

حال به دلیل تاخیرهایی که در انتگرال گیرها، خود ربات و محاسبات مختصات اجماع به وجود می‌آید، از ربات مجازی استفاده شد تا همه ربات‌ها همزمان دستور گیرند. شبیه‌سازی در شکل ۶-۳ آمده است. مکان ربات‌ها در شکل ۷-۳ هستند. همانطور که دیده می‌شود، تاخیر بسیار کم شده است. لذا در ادامه از رهبر مجازی استفاده شده است.



شکل ۳-۶: بلوک‌های شبیه‌سازی شده برای اجماع ربات‌ها با رهبر مجازی



شکل ۳-۷: موقعیت مکانی ربات‌های اجماع با رهبر مجازی

فصل چهارم

مسیریابی

۱-۴ مقدمه

به طراحی مسیری امن به طوری که بدون برخورد با موانع از قرارگیری اولیه، که شامل مکان و زاویه‌ها، به طور کلی تمام پارامترهای توصیف یک جسم در فضای محیط، به قرارگیری نهایی است، مسیریابی^۱ می‌گویند. در این پژوهه می‌توان مکان ربات‌ها و موانع را به عنوان حالت^۲ در نظر گرفت. با توجه به اینکه دید از بالا فرض شده است به نحوی که از تمام مکان‌ها اطلاع داشته باشیم، در نتیجه نسبت به تمام موانع اطلاع دقیقی داریم. پس می‌توان گفت در هر حالت نسبت به محیط و داده‌ها به طور کامل مطلع^۳ هستیم. در این فصل ابتدا روش میدان پتانسیل و سپس الگوریتم‌های مسیریابی مبتنی بر روش‌های ابتکاری^۴ که لازمه آن‌ها مطلع بودن از محیط است، پیاده‌سازی و بررسی گشته‌اند.

۲-۴ میدان پتانسیل

در این روش یک میدان دافعه و جاذبه ایجاد می‌کنیم و ربات در راستای میدان حرکت خواهد کرد. ایده اصلی این است که نقطه‌ی هدف را به صورت جاذبه و نقاطی که مانع هستند را به صورت دافعه در نظر بگیریم (همانند میدان الکتریکی و بارهای همنام و ناهمنام). در نتیجه، ربات به سمت نقطه هدف کشیده خواهد شد.

۱-۲-۴ روابط میدان پتانسیل

رابطه میدان پتانسیل جذبی برای نامین ربات از طرف بار ناهمنام یا به عبارتی موقعیت هدف که یک چاه پتانسیل سه‌می‌گون ایجاد می‌کند، به شرح زیر است [۱۳]:

$$U_{att,i} = \begin{cases} \frac{1}{2}\zeta_i \|o_i(q) - o_i(q_f)\|^2 & \text{if } \|o_i(q) - o_i(q_f)\| \leq d \\ d\zeta_i \|o_i(q) - o_i(q_f)\| - \frac{1}{2}\zeta_i d^2 & \text{otherwise} \end{cases} \quad (1-4)$$

که در آن منظور از (q_i) موقعیت ربات آم، (q_f) موقعیت مطلوب ربات، ζ_i ضریبی است که بعداً در روش گرادیان نزولی به ضریب آن تبدیل می‌گردد و d فاصله گذر از چاه مخروطی را تعیین می‌کند.

¹path planning

²state

³informed

⁴heuristic

علت تعریف میدان به صورت فوض این است که در مرز d هر دو برابر باشند و ناپیوستگی نداشته باشیم.

می‌دانیم $F_{att,i} = -\nabla U_{att,i}$ در نتیجه به توجه به رابطه ۱-۴ خواهیم داشت:

$$F_{att,i} = \begin{cases} -\zeta_i(o_i(q) - o_i(q_f)) & \text{if } \|o_i(q) - o_i(q_f)\| \leq d \\ -d\zeta_i \frac{(o_i(q) - o_i(q_f))}{\|o_i(q) - o_i(q_f)\|} & \text{otherwise} \end{cases} \quad (2-4)$$

برای جلوگیری از برخورد با موانع میدان پتانسیل دفعی به صورت زیر تعریف کرد. این میدان ربات را از موانع دور کرده و اجازه برخورد ربات با مانع را نمی‌دهد. همچنین وقتی ربات از مانع دور است نباید نیرویی از طرف مانع به ربات اعمال گردد. به همین دلیل تابع پتانسیل باید طوری تعریف شود که به هنگام رسیدن ربات به مانع مقدارش بینهایت و در یک فاصله مشخص از مرز مانع صفر گردد. این تابع پتانسیل برای ربات i و مانع j به شکل زیر تعریف می‌گردد:

$$U_{att,ij} = \begin{cases} \frac{1}{2}\eta_{ij}\left(\frac{1}{\rho_j(o_i(q))} - \frac{1}{\rho_{0j}}\right)^2 & \text{if } \rho_j(o_i(q)) \leq \rho_{0j} \\ 0 & \text{otherwise} \end{cases} \quad (3-4)$$

که در آن $(o_i(q))$ و ρ_{0j} به ترتیب برابر یا کوتاهترین فاصله بین زمین ربات و زمین مانع، و فاصله تاثیر زمین مانع هستند. با رابطه ۳-۴ و $F_{rep,ij} = -\nabla U_{rep,ij}$ خواهیم داشت:

$$F_{att,ij} = \begin{cases} \eta_{ij}\left(\frac{1}{\rho_j(o_i(q))} - \frac{1}{\rho_{0j}}\right)\frac{1}{\rho_j^2(o_i(q))}\frac{o_i(q) - o_j(q)}{\|o_i(q) - o_j(q)\|} & \text{if } \rho_j(o_i(q)) \leq \rho_{0j} \\ 0 & \text{otherwise} \end{cases} \quad (4-4)$$

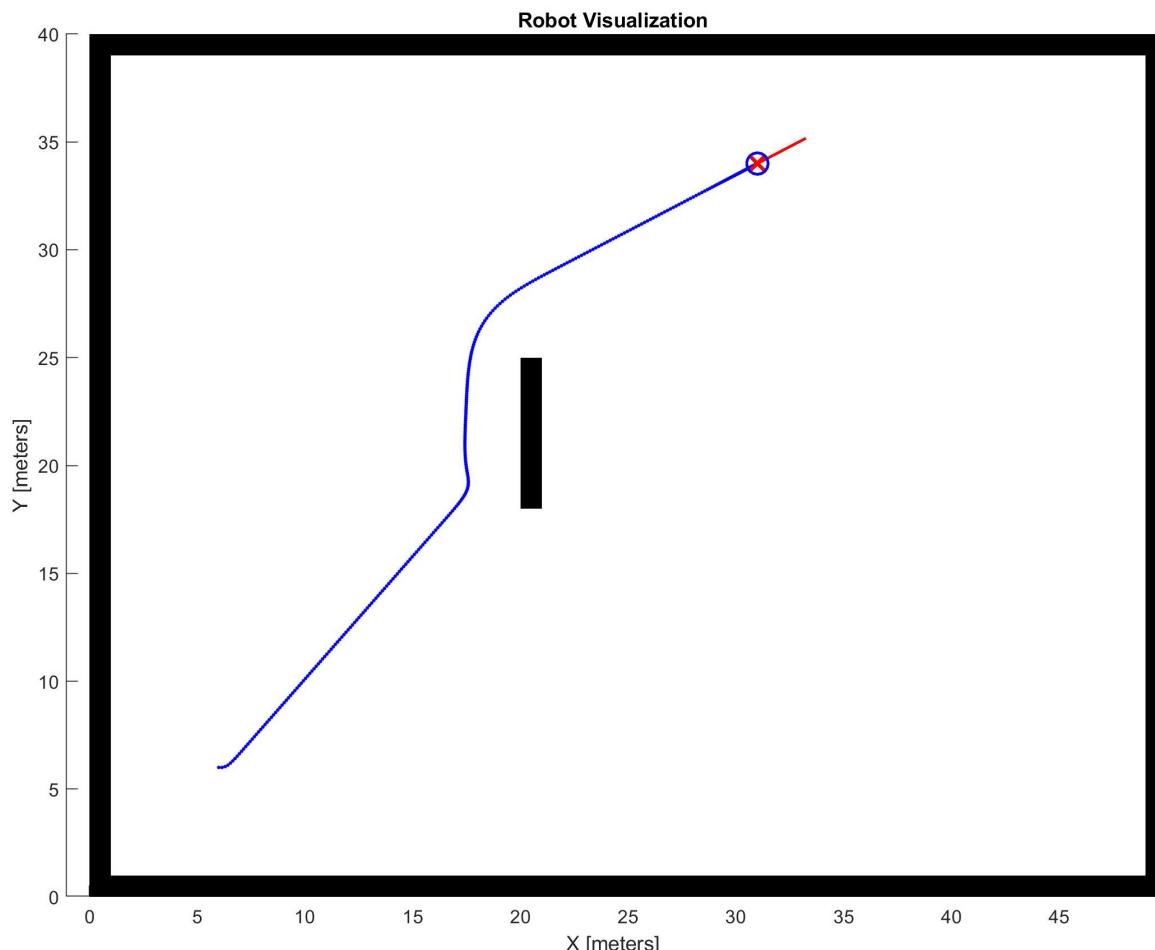
در نهایت نیروی کل اعمالی به ربات به صورت زیر خواهد بود:

$$F_{tot,i} = F_{att,i} + \sum_{j \in obstacles} F_{rep,ij} \quad (5-4)$$

در ادامه نحوه استفاده از نیرو توضیح داده خواهد شد.

۲-۲-۴ کنترل مسیر ربات با میدان پتانسیل

برای این هدف نیروی اعمالی به ربات که در قسمت قبل محاسبه شد را به عنوان سرعت مطلوب به کنترل کننده تک ربات اعمال کردیم تا ربات به سرعت، مسیر خود را تغییر و به جهت مناسب حرکت نماید. مسیر پیموده شده توسط ربات با میدان پتانسیل در شکل ۱-۴ آمده است.



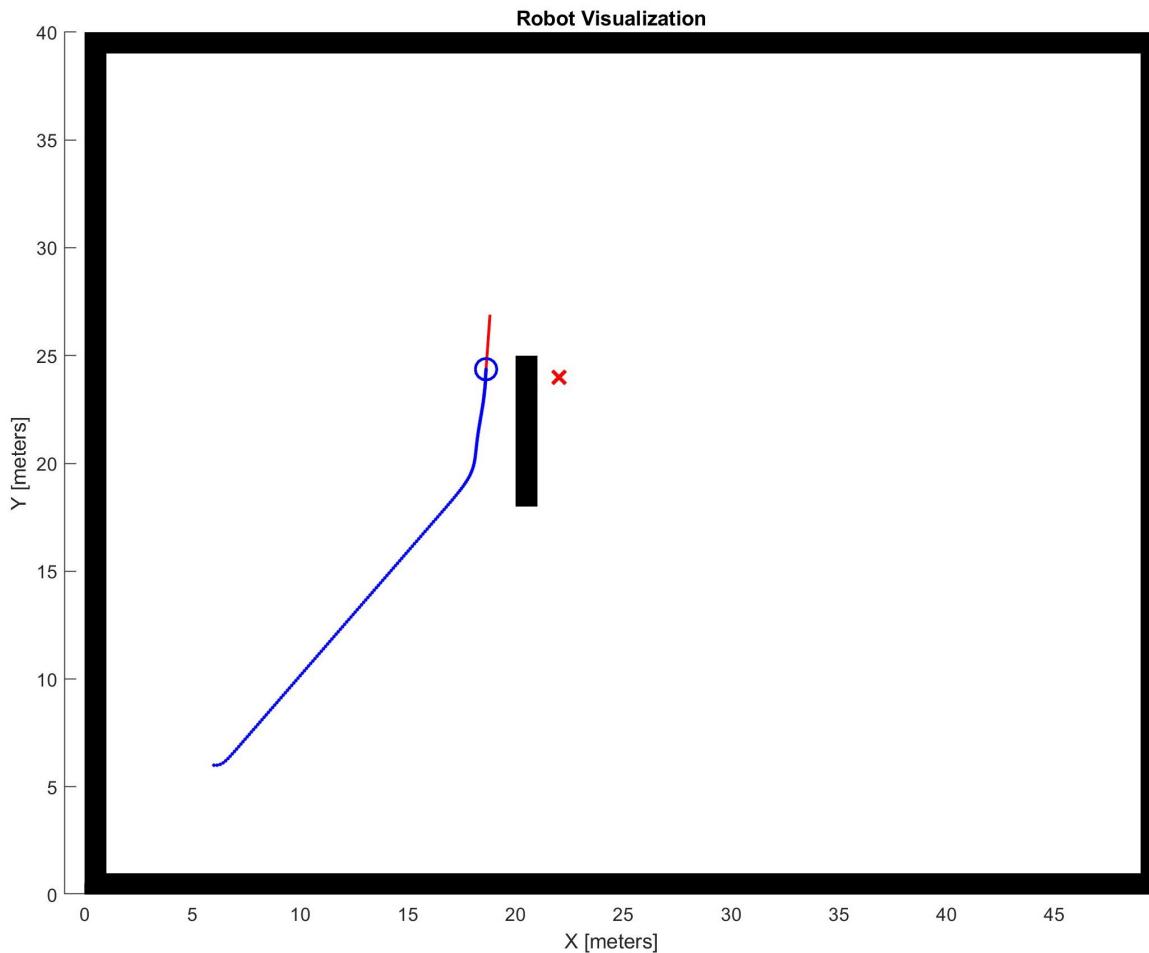
شکل ۱-۴: مسیر پیموده شده توسط ربات با میدان پتانسیل

یکی از مشکلات اصلی میدان پتانسیل این است که لزوماً جواب ندارد و امکان دارد ربات در کمینه^۵ محلی متوقف شود. به عنوان مثال در شکل ۲-۴، ربات در کمینه محلی گیر کرده و به نقطه هدف نرسیده است.

برای حل این مشکل معمولاً نیرویی در جهت تصادفی به ربات وارد می‌گردد اما همچنان تضمینی برای خارج شدن از کمینه محلی^۶ وجود ندارد. لذا در ادامه به بررسی الگوریتم‌هایی می‌پردازیم که مطمئن

⁵minimum

⁶local minimum



شکل ۲-۴: گیر کردن ربات در کمینه محلی با میدان پتانسیل

هستیم حتماً مسیری که به موقعیت مطلوب ختم می‌شود را می‌یابند.

۳-۴ الگوریتم‌های ابتکاری

همانطور که گفته شد، با توجه به شرایط مسئله و مطلع بودن در زمان طراحی مسیر، از الگوریتم‌های ابتکاری استفاده شده است. خصوصیت این الگوریتم‌ها در این است که با ابتکاری که به خرج داده می‌شود، مسئله اندکی ساده‌تر فرض شده و حل می‌گردد. مزیت این الگوریتم‌ها پیچیدگی زمانی کمتر آن‌ها نسبت به مسائلی هستند که مسئله مسیریابی را به طور کامل حل می‌نمایند. این امر در کاربرد رباتیک به دلیل اهمیت بلادرنگ^۷ بودن، دارای اهمیت بسیار زیادی می‌باشد. اما همانطور که گفته شد،

⁷real time

چون مسئله به طور کامل حل نمی‌گردد، تضمینی برای رسیدن به پاسخ بهینه^۸ وجود ندارد اما پاسخ به طور کلی معقول است، هر چند بهینه نباشد.

برای اعمال این الگوریتم‌ها نیاز به داشتن تعداد حالات محدود بود اما در بازه بین دو عدد متمایز در اعداد حقیقی، مجموعه حالات نامتناهی هستند. لذا از روش نمونه‌برداری^۹ استفاده شد. در این پروژه، تمام نقاط با مختصات‌های اعداد صحیح را به عنوان حالت‌ها در نظر گرفتیم. همچنین مجموعه اقدامات^{۱۰} به صورت حرکت به حالت‌های مجاور در راستاهای شمال، جنوب، شرق و غرب، و همچنین در راستاهای مورب شمال-غرب، شمال-شرق، جنوب-غرب و جنوب-شرق، تعریف شدند. مسیر خروجی نیز به صورت مجموعه‌ای مرتب از مختصات‌های مجاور که شروع آن مختصات اولیه و پایان آن مختصات نهایی می‌باشد، است. در این پروژه روش‌های ابتکاری^{۱۱} BFS، Greedy best-first search^{۱۲} و A^* پیاده سازی شده‌اند که در ادامه به توضیح آن‌ها می‌پردازیم.

۱-۳-۴ الگوریتم BFS

الگوریتم BFS در اصل یکی از الگوریتم‌های سرج معروف و پایه‌ای گراف می‌باشد. برای استفاده از این روش، گراف را به نحوی که هر راس نماینده فقط و فقط یک حالت مجاز، و همچنین هر اقدام نماینده فقط و فقط یک یال باشد ساختیم. همچنین وزن هر یال برابر با نرم^{۱۳}-۲ بوده‌است. جایی اقدام مربوطه، که همان فاصله در مختصات دکارتی است، در نظر گرفته شد. منظور از حالت مجاز، مختصاتی است که در آن مانع نباشد و در محدوده کاری قرار گیرد. حال الگوریتم BFS بر روی گراف اعمال می‌شود. قابل ذکر است در صورتی که وزن تمام یال‌ها برابر باشند، اثبات می‌گردد که این الگوریتم به جواب بهین خواهد رسید.

نحوه کار این الگوریتم به این صورت است که با شروع از راس شروع، راس‌های مجاور بررسی می‌شوند و سپس وارد یک صف^{۱۴} می‌گردند تا راس‌های مجاور آن‌ها بررسی گردد. صف در ابتدا تنها دارای راس شروع است. برای اینکه راس‌های تکراری وارد صف نگرددند، از رنگ‌آمیزی راسی استفاده شد. به این صورت که در ابتدا همه راس‌ها سفید و تنها راس شروع، مشکی می‌باشد. سپس هر راسی که وارد صف

⁸optimal

⁹sampling based method

¹⁰actions

¹¹Breadth-first search

¹²2-norm

¹³queue

می‌شود به رنگ مشکی در می‌آید. در نتیجه هنگام بررسی راس‌های مجاور ابتدا بررسی می‌گردد که چه رنگی دارد. زیرا اگر مشکی باشد یعنی قبلاً بررسی شده و دیگر نیازی به بررسی نیست. اینکار تا زمانی که به راس هدف برسیم، ادامه خواهد یافت.

اگر گراف به صورت $G = (V, E)$ که در آن V مجموعه راس‌ها و E مجموعه یال‌ها می‌باشد، تعریف شود، الگوریتم اصلی BFS در شکل ۳-۴ به صورت شبه کد آمده است [۴]. منظور از نماد Q در شبه کد، صف، منظور از d فاصله و منظور از π والد^{۱۴} است. تفاوت در کد ارائه شده در این پروژه با این شبه کد، این است که احتیاجی به رنگ خاکستری نبود و رنگ خاکستری معادل رنگ مشکی در نظر گرفته شد. همچنین چون نقطه هدف داشتیم، در صورتی که به نقطه هدف می‌رسیدیم، حلقه خط ۱۰ شکسته می‌شد و الگوریتم متوقف می‌گشت.

$BFS(G, s)$

```

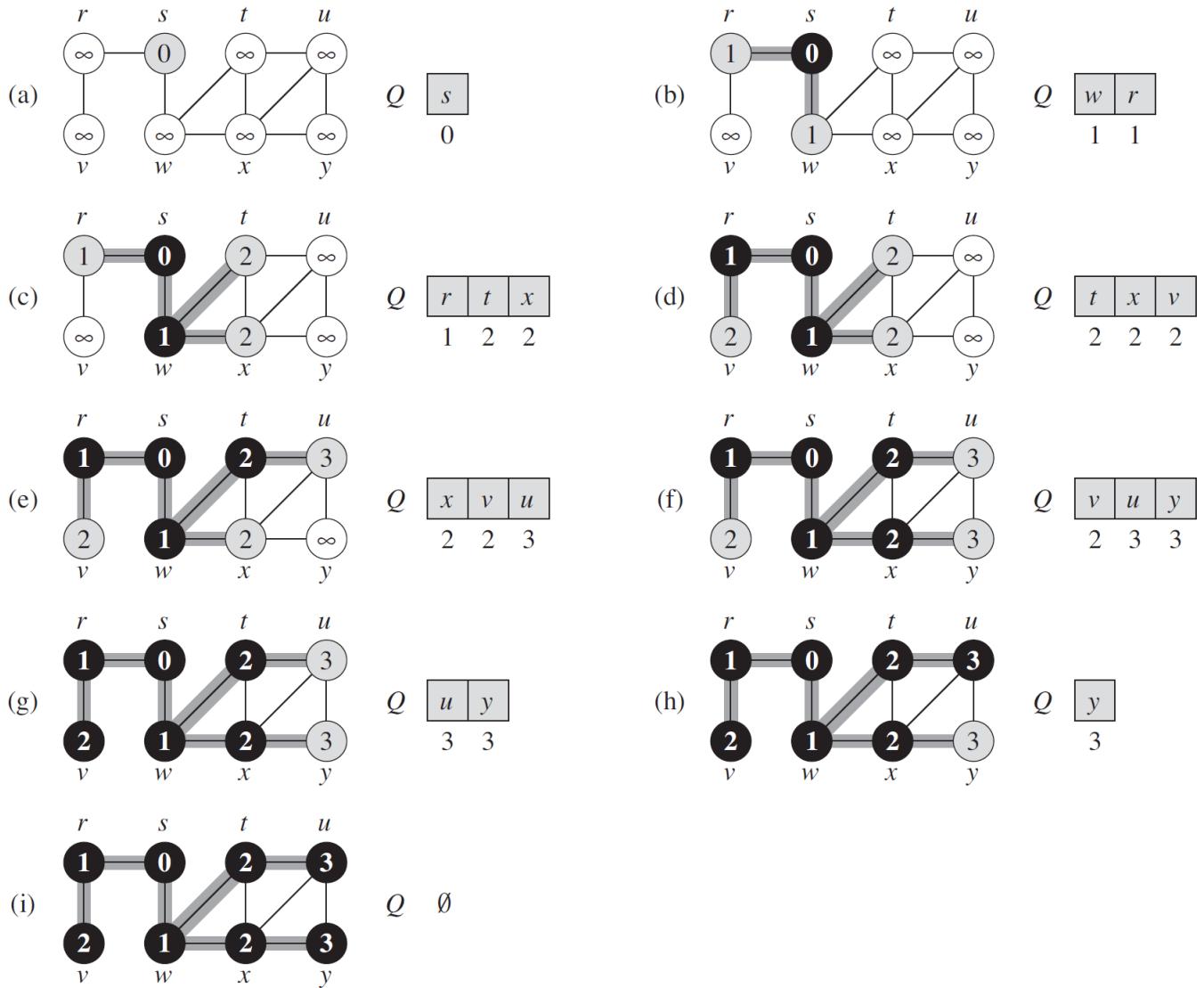
1  for each vertex  $u \in G.V - \{s\}$ 
2     $u.color = \text{WHITE}$ 
3     $u.d = \infty$ 
4     $u.\pi = \text{NIL}$ 
5   $s.color = \text{GRAY}$ 
6   $s.d = 0$ 
7   $s.\pi = \text{NIL}$ 
8   $Q = \emptyset$ 
9  ENQUEUE( $Q, s$ )
10 while  $Q \neq \emptyset$ 
11    $u = \text{DEQUEUE}(Q)$ 
12   for each  $v \in G.Adj[u]$ 
13     if  $v.color == \text{WHITE}$ 
14        $v.color = \text{GRAY}$ 
15        $v.d = u.d + 1$ 
16        $v.\pi = u$ 
17       ENQUEUE( $Q, v$ )
18    $u.color = \text{BLACK}$ 

```

شکل ۳-۴: شبکه کد الگوریتم BFS

در شکل ۴-۴ نیز نحوه اعمال شدن الگوریتم BFS بر روی یک گراف، نمایش داده شده است.

^{۱۴}parent


 شکل ۴-۴: نحوه اعمال شدن الگوریتم BFS

۲-۳-۴ الگوریتم BFS

ابتکار این روش این است که سعی می‌کند به سمت راسی که کمترین فاصله تا راس هدف دارد، گسترش پیدا کند، به امید آنکه به کوتاه‌ترین مسیر دست یابد [۱۰]. چون کمترین فاصله تا راس هدف را نداریم، حد پایین آن را تقریب می‌زنیم و برابر فاصله قرار می‌دهیم. برای تقریب حد پایین، چون مختصات دو راس را داریم، کافی است طول خط راستی که راس را به راس هدف می‌رساند به دست آورد. در این راستا تابع $h(n)$ برای مختصات راس، n ، تعریف می‌گردد. اگر مختصات راس هدف با n' نمایش داده

شود، مقدار تابع $h(n)$ که طول خط راست از راس تا راس هدف است به صورت زیر تعریف می‌گردد:

$$h(n) = \|n - n'\|_2 \quad (6-4)$$

مشخصاً این روش نیز لزوماً به جواب بهین نمی‌رسد، زیرا اصلاً می‌تواند بین راسی که انتخاب شده تا راس هدف مسیری نباشد و به بن بست بررسیم و مجبور به برگشت باشیم. همچنین برای مثال نقض نیز می‌توان نتیجه این روش را در بخش ۴-۳-۴ بررسی کرد که جواب آن بهین نیست.
در هنگام جست‌وجو در هر گام، والد هر راس را در درخت مسیر، به روز می‌گردد و در انتهای با کمک از همین والدها و با شروع از راس هدف، تا راس شروع، به عقب برمنی‌گردیم. بر عکس این مسیری، مسیری خواهد بود که ربات باید بپیماید.

۴-۳-۴ الگوریتم A^*

الگوریتم A^* ادامه‌ای بر الگوریتم Greedy best-first search است که در آن علاوه بر تابع $h(n)$ که در رابطه ۶-۴ تعریف شد، تابع $g(n)$ نیز تعریف می‌شود که مسافت واقعی بین نقطه شروع تا راس n است. این فواصل با الگوریتم جست‌وجو مرتباً به روز می‌شود و در صورتی که مسیر کوتاهتری یافته شود نیز مقادیر آن به همراه مسیر واصله، به روز خواهند شد. همچنین برای یافتن کوتاهترین مسیر نیز همانند الگوریتم Greedy best-first search، برای هر راس یک والد در نظر گرفته شده است و با پیدا کردن مسیری کوتاهتر، مقداری کمتر برای تابع $g(n)$ در راس n ، والد نیز به راس جدیدی که به آن مسیر دارد، تغییر می‌یابد [۱۰].

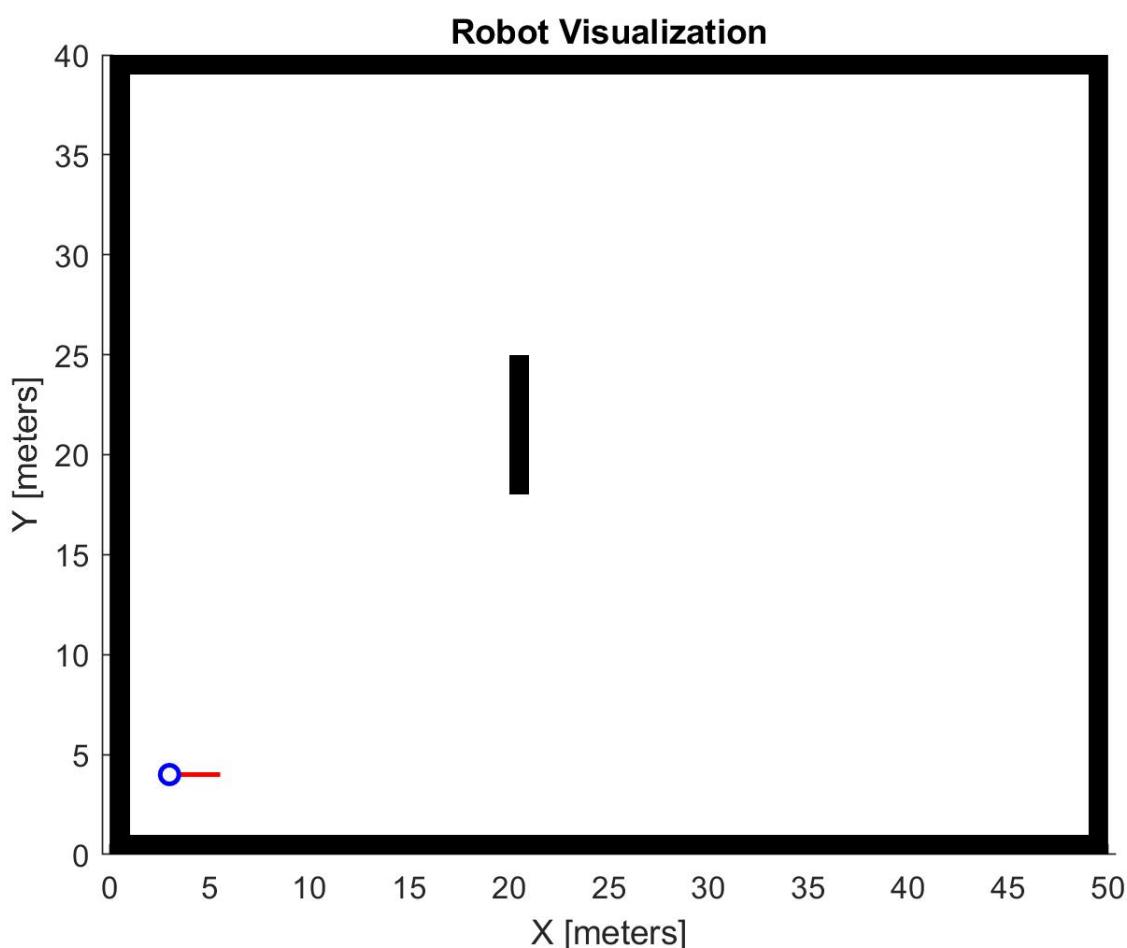
تابعی که طبق آن، راسی که کمترین مقدار آن را داشته باشد جست‌وجو خواهد شد، $f(n)$ می‌نامیم، که به صورت زیر تعریف می‌گردد:

$$f(n) = g(n) + h(n) \quad (7-4)$$

در نهایت نیز هنگامی که به راس هدف بررسیم، الگوریتم جست‌وجو متوقف می‌گردد و سپس به طور بازگشتی از راس هدف، بر روی والدها به عقب برمنی‌گردیم تا با عکس کردن جهت حرکت، مسیر ربات تعیین شود.

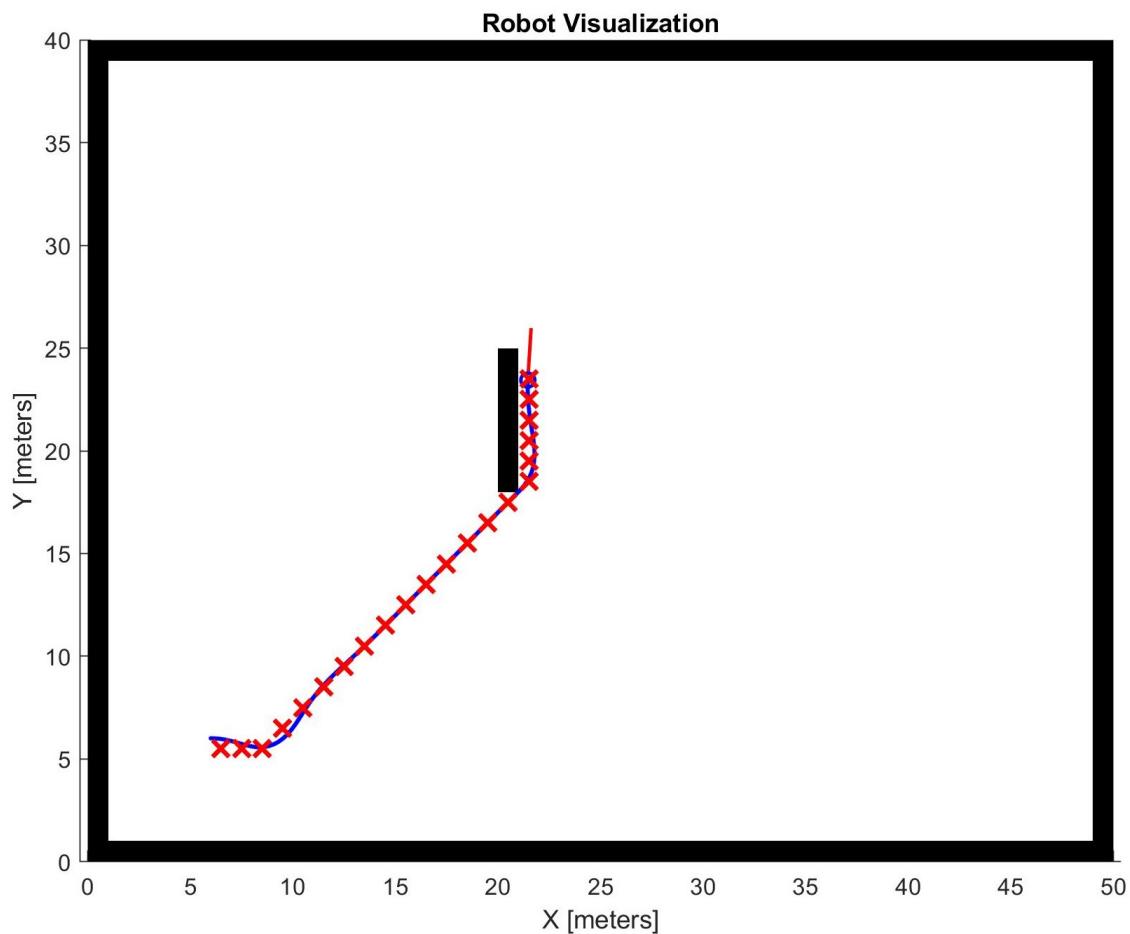
۴-۳-۴ نتایج و مقایسه الگوریتم‌های ابتکاری

در این قسمت ابتدا به بررسی مسیر یافته شده توسط الگوریتم‌ها پرداخته می‌شود و سپس الگوریتم‌ها از نظر زمانی و ابعاد نقشه، مورد بررسی قرار می‌گیرند. نقشه مورد استفاده برای بررسی پاسخ‌ها در شکل ۵-۴ آمده است. نقاط مشکی رنگ، مکانی هستند که مانع وجود دارد.



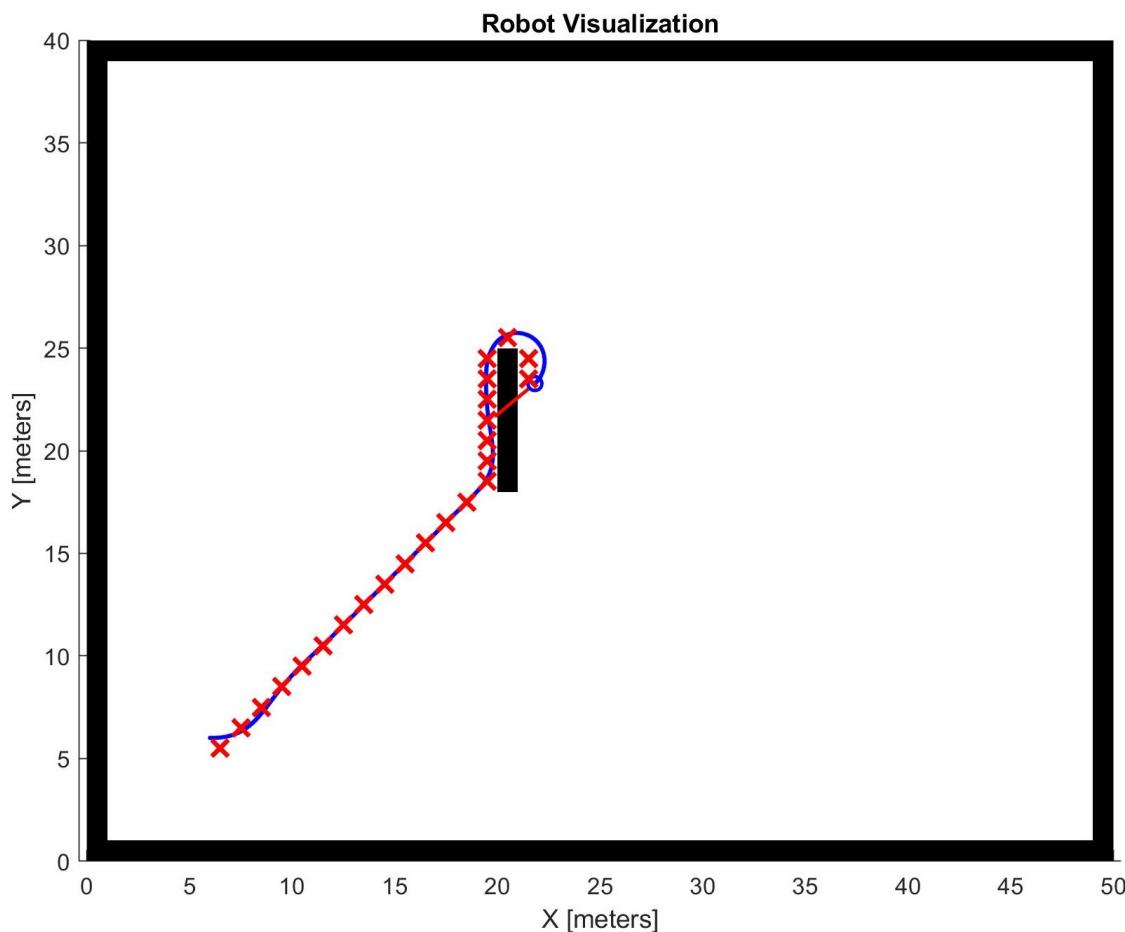
شکل ۴-۵: نقشه‌ای که الگوریتم‌ها به آن اعمال شده‌اند

حال برای هر سه الگوریتم، مسیری برای نقطه شروع $[6 \ 24]^T$ و هدف $[21 \ 6]^T$ خواسته شد. نتایج به شکل‌های ۴-۶، ۷-۴ و ۸-۴ آمده است. نقاط ضربدر خورده قرمز، مسیر الگوریتم، و خط آبی مسیر پیموده شده توسط ربات هستند.



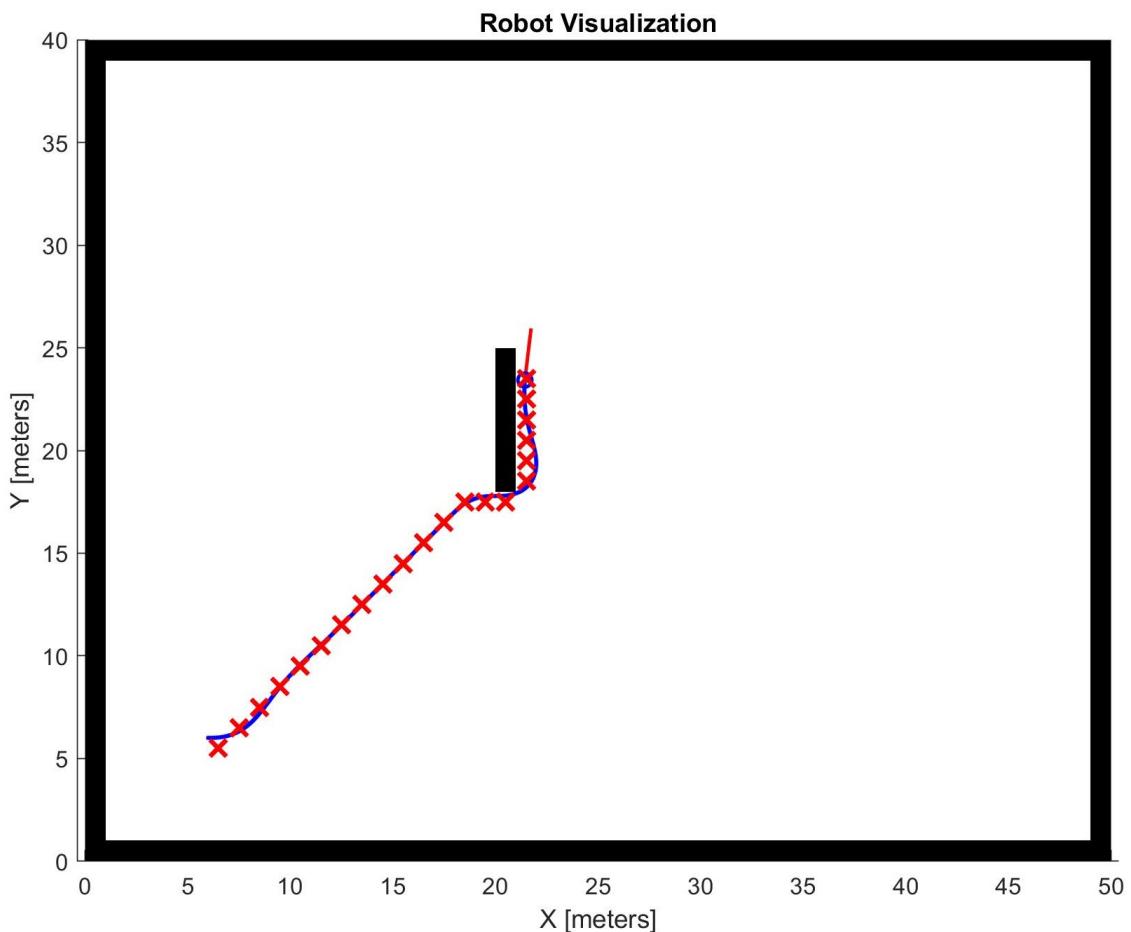
شکل ۴-۶: مسیر پیشنهادی توسط الگوریتم BFS

همانطور که می‌شود دید در این مثال، الگوریتم Greedy best-first search، مسیر طولانی‌تری نسبت به دو الگوریتم دیگر ارائه داده است. همچنین مسیرهای ارائه شده توسط دو الگوریتم، A^* و BFS، هر دو بهین هستند.



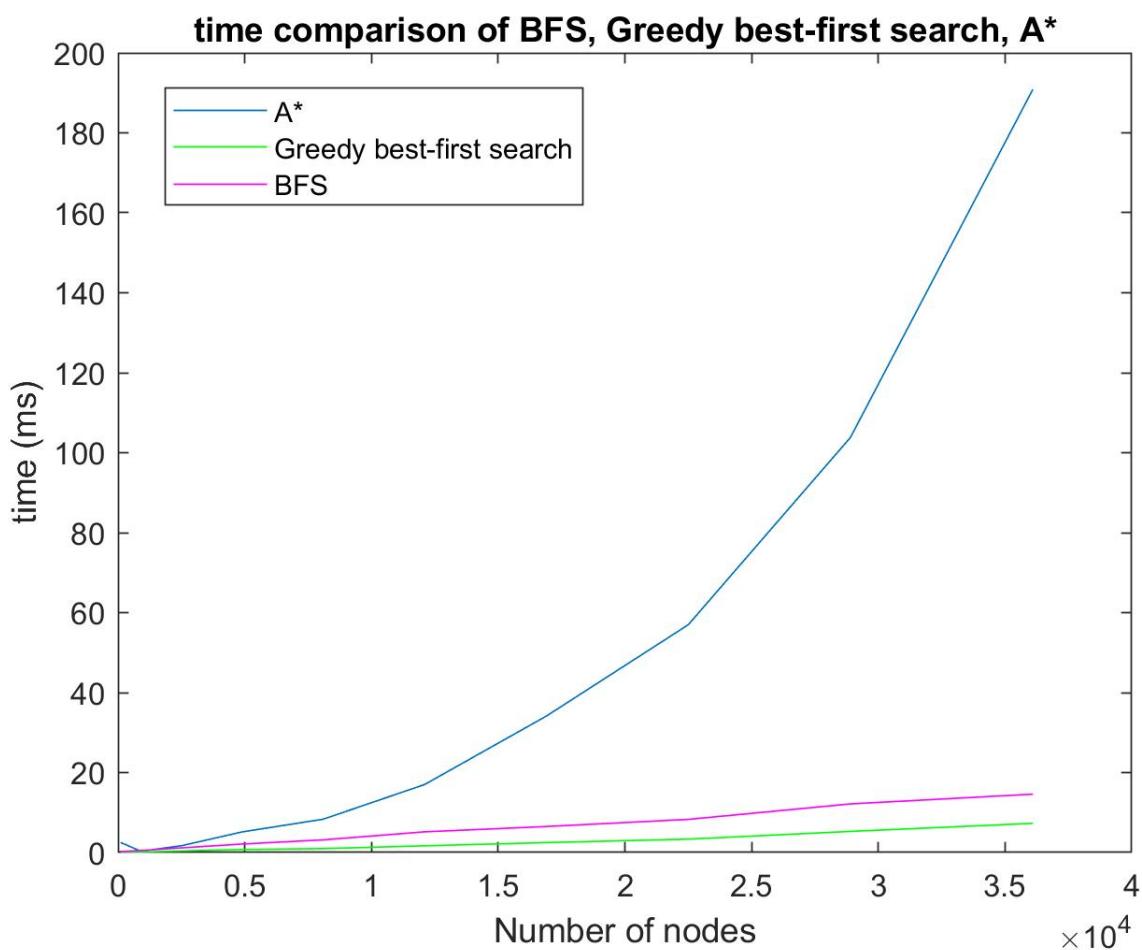
شکل ۷-۴: مسیر پیشنهادی توسط الگوریتم Greedy best-first search

نکته دیگر قابل توجه، این است که هنگام چرخش به خصوص چرخش‌های سنگین همانند برگشت ربات در شکل ۷-۴ در بالای موانع، فشار زیادی به ربات آمده و کنترل مسیر برای او سخت بوده است. در نتیجه بهتر است کنترل کننده بهتری برای ربات در نظر گرفته شود. همچنین استفاده از الگوریتم‌هایی که سعی کنند میزان فشار واردہ بر ربات را در تصمیمشان در نظر گرفته و کاهش دهنند نیز، می‌تواند عضو کارهای آینده باشد.



شکل ۴-۸: مسیر پیشنهادی توسط الگوریتم A^*

حال نوبت بررسی زمان مورد نیاز برای اجرای هر الگوریتم است. در این راستا، سه الگوریتم با نقاط تصادفی شروع و هدف، در حالی که برای هر سه یکسان باشند، اجرا گشتند. به ازای هر اندازه، ۱۰۰ بار الگوریتم‌ها برای نقاط متفاوت بررسی و سپس از زمان آن‌ها میانگین گرفته شد. نتیجه در شکل ۹-۴ به نمایش در آمده است. همانطور که مشاهده می‌شود، زمان الگوریتم A^* به مراتب بیشتر از الگوریتم‌های دیگر است. همچنین الگوریتم Greedy best-first search کمترین زمان را دارد. در نتیجه هر چقدر بخواهیم به جواب معقولتری بررسیم، مدت زمان بیشتر مورد نیاز است. در حالت کلی با توجه به اینکه برای ۳۵۰۰۰ نود، در کمتر از ۲۰۰ میلی ثانیه برای الگوریتم A^* به جواب رسیده است، استفاده از این الگوریتم توصیه می‌شود. قابل ذکر است که الگوریتم A^* روش معقول‌تری نسبت به دو الگوریتم دیگر دارد و همچنین عملکرد آن در شرایطی که وزن یال‌ها نسبت به هم بسیار متفاوت باشند بهتر نیز خواهد بود و قابلیت تعیین به مسائل دیگر را نیز دارد.



شکل ۴-۹: مقایسه زمانی سه الگوریتم A^* و Greedy best-first search، BFS

Q-learning ۴-۴

Q-learning یک روش بدون مدل یادگیری تقویتی^{۱۵} است. همچنین می‌توان به آن به دید یک روش آسنکرون^{۱۶} برنامه ریزی داینامیک^{۱۷} اشاره کرد [۱۶]. محیط را همانند شرایط قبل نسبت به آن اطلاع کامل داریم. همچنین اقدامات ما نتایج قطعی^{۱۸} دارند. حال به تعریف مجدد حالتها و اقدامات می‌پردازیم. برای سادهسازی تمام مکان‌های عدد صحیح در محدوده کاری خود را به عنوان یک حالت در نظر گرفتیم.

¹⁵reinforcement learning

¹⁶asynchronous

¹⁷dynamic programming (DP)

¹⁸deterministic

مجموعه A که مجموعه اقدامات است به شرح زیر تعریف می‌شوند:

$$A = \{[0 \ 1]^T, [0 \ -1]^T, [1 \ 0]^T, [-1 \ 0]^T, [1 \ 1]^T, [-1 \ 1]^T, [1 \ -1]^T, [-1 \ -1]^T\} \quad (8-4)$$

تنها نگته قابل توجه این است که در کناره‌های نقشه اگر از نقشه خارج می‌شدیم باید حالت بعدی را حذف نماییم.

هدف کلی این روش دست یافتن به سیاست بهینه^{۱۹} از روی تابع Q و ارزش هر حالت، V طبق پاداش^{۲۰} گرفته شده، است [۱۵]. در ادامه به توضیح الگوریتم و روابط آن پرداخته می‌شود.

۱-۴-۴ الگوریتم Q-learning

با توجه به [۱۶] تمام روابط برای حالت کلی‌تر که محیط غیرقطعی باشد نوشته شده است. لذا برای هر انتقال^{۲۱} یک احتمال در نظر گرفته شده است. در مسئله ما به دلیل اینکه محیط قطعی است، احتمال برای یک حالت ۱ و بقیه حالت‌ها صفر خواهد شد. در نتیجه از نوشتن آن خودداری شده است. ارزش هر حالت بر طبق سیاست π به صورت زیر تعریف می‌گردد:

$$V^\pi(x) \equiv R_x(\pi(x)) + \gamma V^\pi(y) \quad (9-4)$$

که در آن y حالتی است که از حالت x طبق سیاست π به آن خواهیم رسید. همچنین نسبت ارزشی که s گام قبل تر گذاشته شده نسبت به ارزش الان، γ^s برابر است ($0 < \gamma < 1$). در نتیجه برای سیاست بهینه π^* خواهیم داشت:

$$V^*(x) \equiv V^{\pi^*}(x) = \max_a \{R_x(a) + \gamma V^{\pi^*}(y)\} \quad (10-4)$$

حال عبارتی که باید از آن نسبت به هر اقدام، ماکزیمم گرفته شود را تابع Q تعریف می‌کنند.

$$Q^\pi(x, a) = R_x(a) + \gamma V^\pi(y) \quad (11-4)$$

¹⁹optimal policy

²⁰reward

²¹transaction

حال در گام n ، هر حالت و اقدام مطابق روابط $12-4$ و $13-4$ توابع Q و V به روزرسانی می‌گردد.

$$Q_n(x, a) = (1 - \alpha_n)Q_{n-1}(x, a) + \alpha_n[r_n + \gamma V_{n-1}(y_n)] \quad (12-4)$$

که در آن داریم:

$$V_{n-1}(y) \equiv \max_b\{Q_{n-1}(y, b)\} \quad (13-4)$$

۲-۴-۴ پیاده‌سازی الگوریتم Q-learning

به دو روش متفاوت الگوریتم Q-learning پیاده‌سازی و در [۲] قابل دسترسی است. در روش اول از برنامه ریزی دینامیک استفاده نشده است و با شروع از مکان اولیه ربات، به صورت رندوم، به حالت‌های مجاور رفته و به روز رسانی انجام شده است. در روش دوم از برنامه ریزی دینامیک استفاده شده است. چون در برنامه ریزی دینامیک هدف این است که با اندکی افزودن به پیچیدگی حافظه استفاده شده، از پیچیدگی زیاد زمانی بپرهیزیم، در روش اول، محاسبات بسیار طولانی بودند، به نحوی که حتی پاسخ‌ها هم تفاوت زیادی تا پاسخ بهینه داشتند. اما در روش دوم، به صورت گام به گام، هر بار تمام حالت‌ها و اقدامات بررسی و به روزرسانی می‌شدند و با بررسی بیشترین تغییر، اگر از حد تعیین شده کمتر می‌شد الگوریتم متوقف می‌گشت. لذا دید خوبی نیز نسبت به شرایط الگوریتم و مقدار بهینگی به دست می‌آمد و از محاسبات اضافی جلوگیری می‌شد.

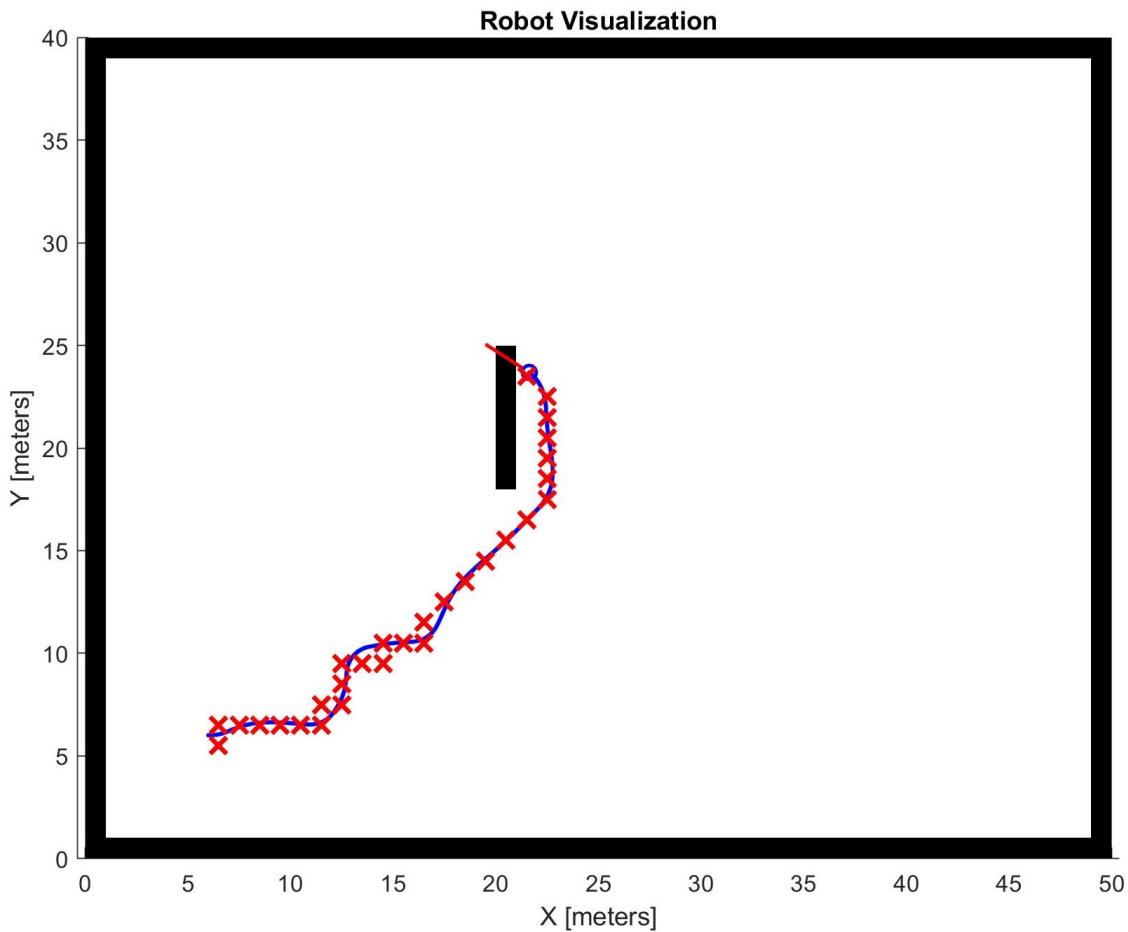
نکته مهم در این الگوریتم، تنظیم کردن 22 مقدار α و γ و همچنین تعیین یک پاداش مناسب r می‌باشد. در بخش بعد به خصوص در مورد انتخاب‌های پاداش، صحبت خواهد شد.

۳-۴-۴ نتایج مسیریابی با Q-learning

در ابتداء نتیجه روش‌های اول و دوم بررسی شده و سپس بعد از مشخص شدن بررسی روش دوم، پاداش‌های متفاوت برای روش دوم بررسی خواهند شد. نتیجه روش اول برای یک میلیون بار شروع و رسیدن به نقطه هدف یا مانع در شکل $10-4$ آمده است.

همانطور که مشخص است زمان مورد نیاز برای یک میلیون پردازش بسیار زیاد است و یک ضعف

²²tuning

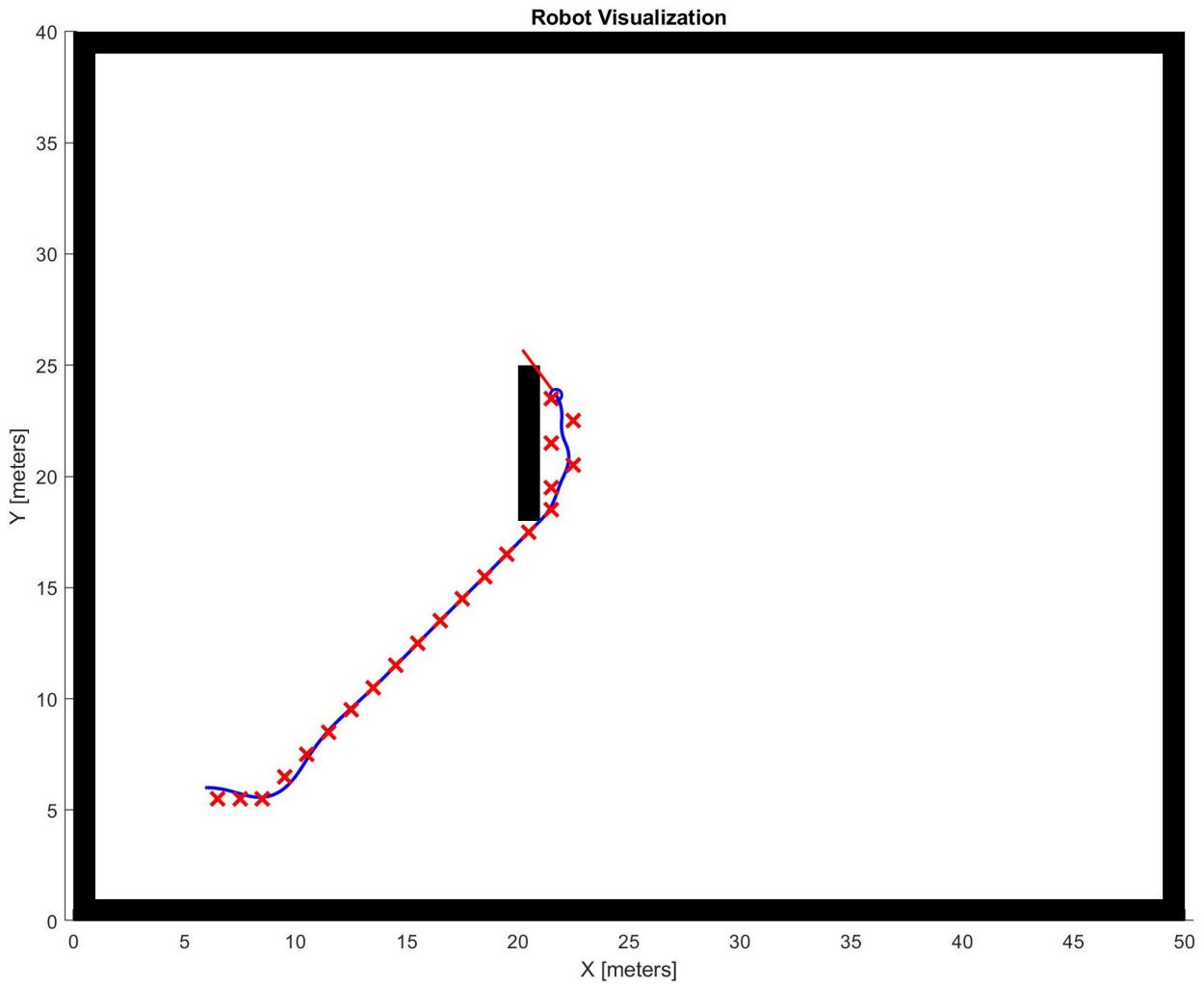


شکل ۴-۱۰: نتیجه الگوریتم Q-learning برای یادگیری تصادفی

دیگر این الگوریتم یادگیری این است که بسیار طول می‌کشد تا به پاسخ صحیح دست یابد. با روش برنامه‌ریزی داینامیک برای الگوریتم یادگیری، نتایج بسیار بهتری از نظر زمانی و بهینگانی به دست آمد. همچنین این الگوریتم سیاست بهینه را برای تمامی نقاط محاسبه می‌کند. در ادامه این پاسخ‌ها بررسی خواهند شد.

همانطور که گفته شد، با داشتن بیشترین تغییر در مقادیر V یک معیار خوب برای درجه یادگیری الگوریتم وجود دارد. حال توابع مختلف پاداش را بررسی می‌گردد. در همه توابع اگر مدل به حالت هدف برسد با مقدار $R = 10$ پاداش می‌گیرد و اگر به موانع بخورد با از صفحه خارج شود به مقدار $R = -10$ جریمه می‌شود. در شکل ۱۱-۴ نتیجه مسیریابی با $R = -1$ به ازای بقیه حالت‌ها نشان داده شده است. دلیل این مقدار دهی این است که ربات سعی کند با کمترین اقدام به حالت هدف برسد. این مسیریابی بعد از ۷۶ گام به نتیجه رسیده است.

همانطور که دیده می‌شود در انتهای مسیریابی زیگ زاگ شده است. دلیل این مسئله این است که اگر

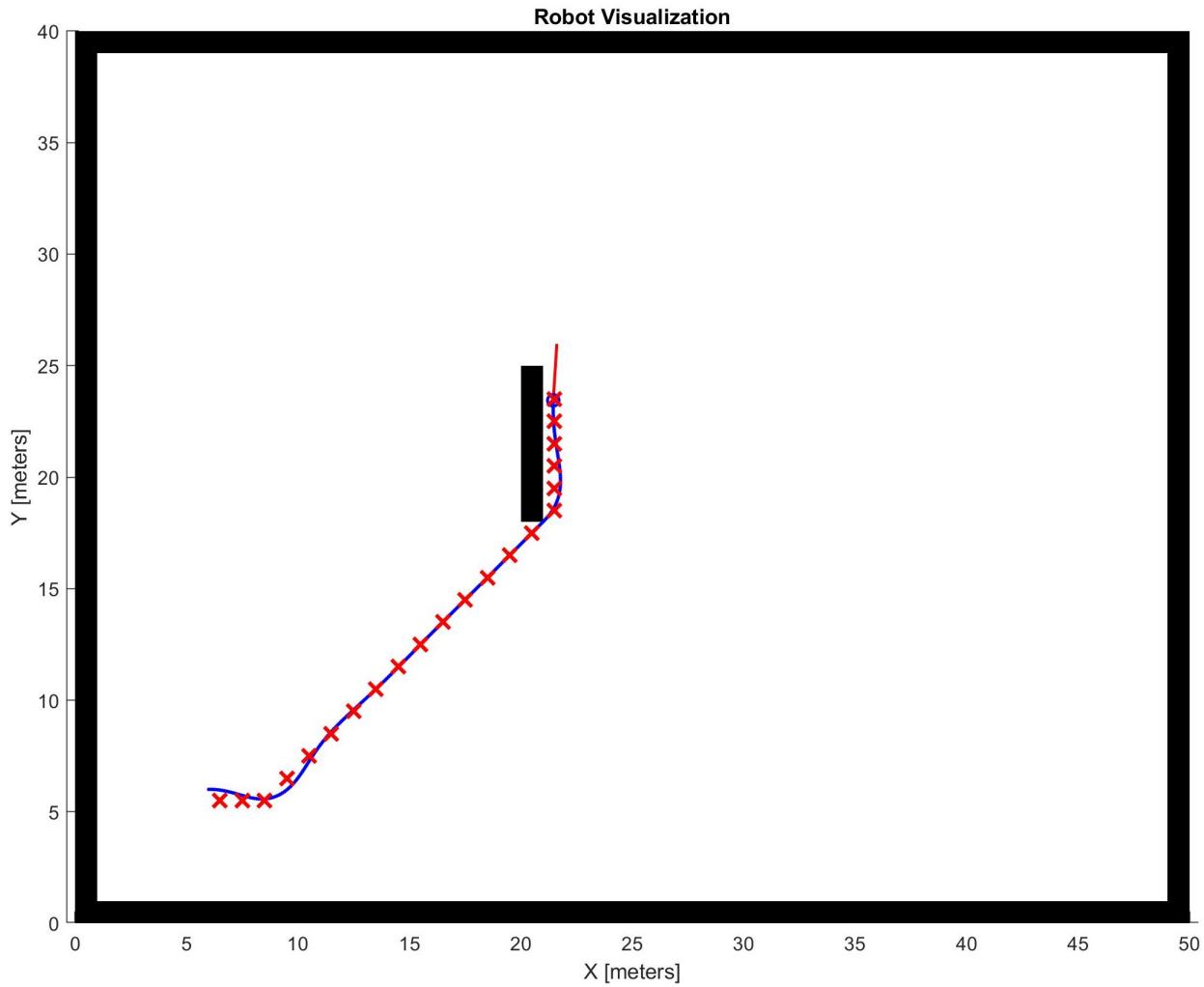


شکل ۱۱-۴: نتیجه الگوریتم Q-learning برای یادگیری با $R = -1$

ربات به صورت مستقیم حرکت کند یا زیگ زاگ، از نظر تعداد اقدام‌ها جریمه‌ای متفاوت دریافت نخواهد کرد. برای اینکه ربات تفاوت میان حرکت مستقیم یا زیگ زاگ را درک کند باید توجه کنیم که هدف ما تعداد اقدامات نیست بلکه مسافت پیموده شده است. لذا طبق رابطه ۸-۴ برای هر اقدام جریمه‌ای برابر با $R = -\|a\|_2$ در نظر می‌گیریم که برابر با مسافت پیموده شده تحت هر اقدام است. در نتیجه هدف مینیمم کردن مسیر پیموده شده است. نتیجه این روش در شکل ۱۲-۴ بعد از ۷۹ گام به نمایش در آمده است.

در نتیجه مقداردهی پاداش و جریمه به صورت

$$R = -\|a\|_2 \quad (14-4)$$



شکل ۱۲-۴: نتیجه الگوریتم Q-learning برای یادگیری با $R = -\|a\|_2$

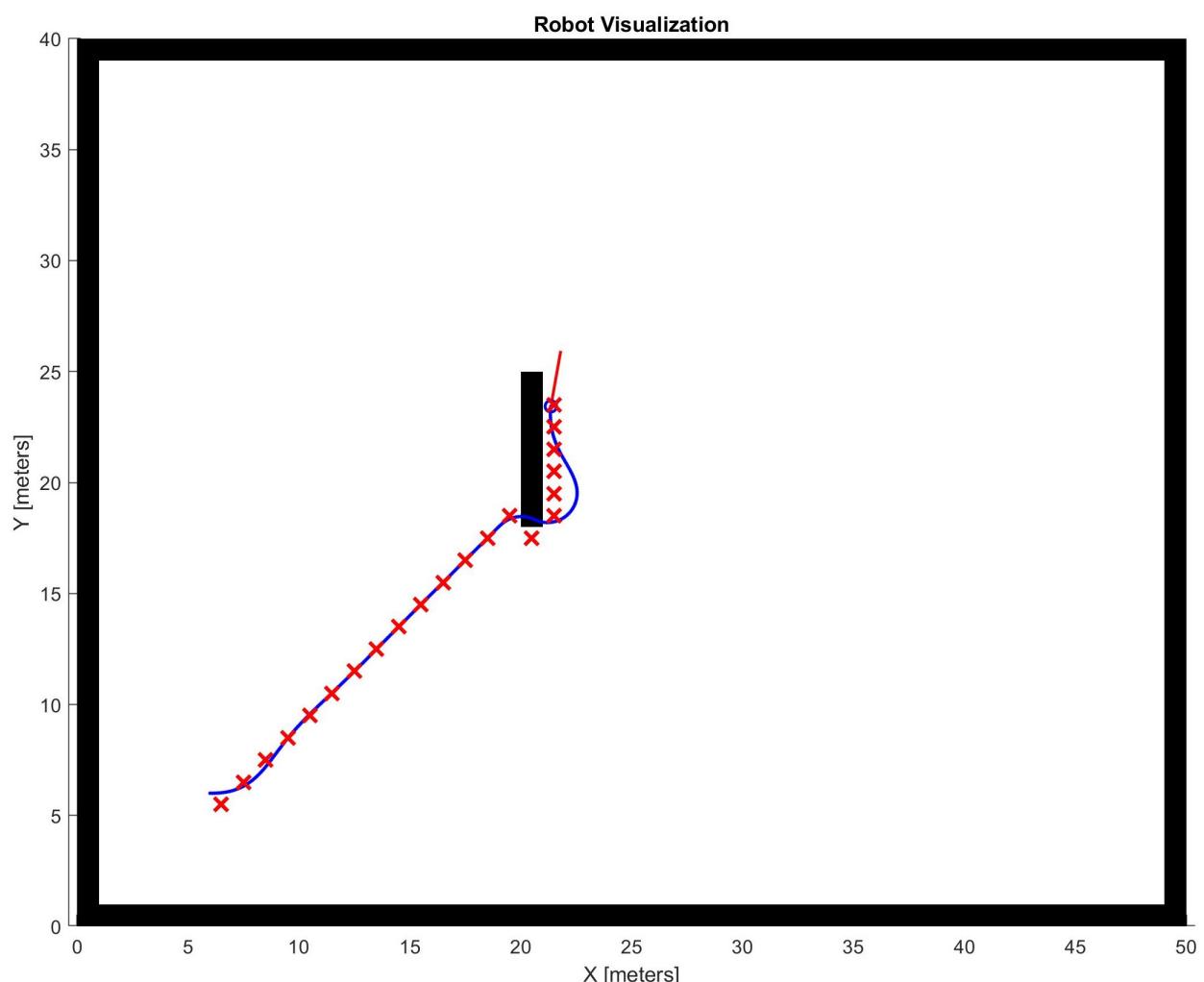
بهترین روش انتخاب است و با زمان کافی به پاسخ بهین خواهد رسید. در ادامه پروژه نیز از آن استفاده خواهد شد. حال به تعدادی از اشتباهات ممکن در مقداردهی R پرداخته خواهد شد.

۴-۴-۴ اشتباهات رایج در انتخاب پاداش

در ابتدا مفهوم این مطلب در بازی شطرنج مثال زده خواهد شد و سپس به مسئله این پایان نامه پرداخته خواهد شد.

اگر برای یادگیری شطرنج توسط یک مدل هدف نهایی که مات کردن حریف و پیروزی است را پاداش بگیریم اما در کنارش به بردن مهره‌ها هم امتیاز دهیم، امکان دارد مدل به جای تمرکز بر پیروزی بیشتر دنبال گرفتن مهره‌های حریف باشد مخصوصاً اگر پاداش پیروزی به نسبت گرفتن مهره‌ها بسیار بزرگ نباشد و گرفتن مهره نسبت به پیروزی برای او اهمیت بیشتری داشته باشد. یا حتی تنها پیروزی را

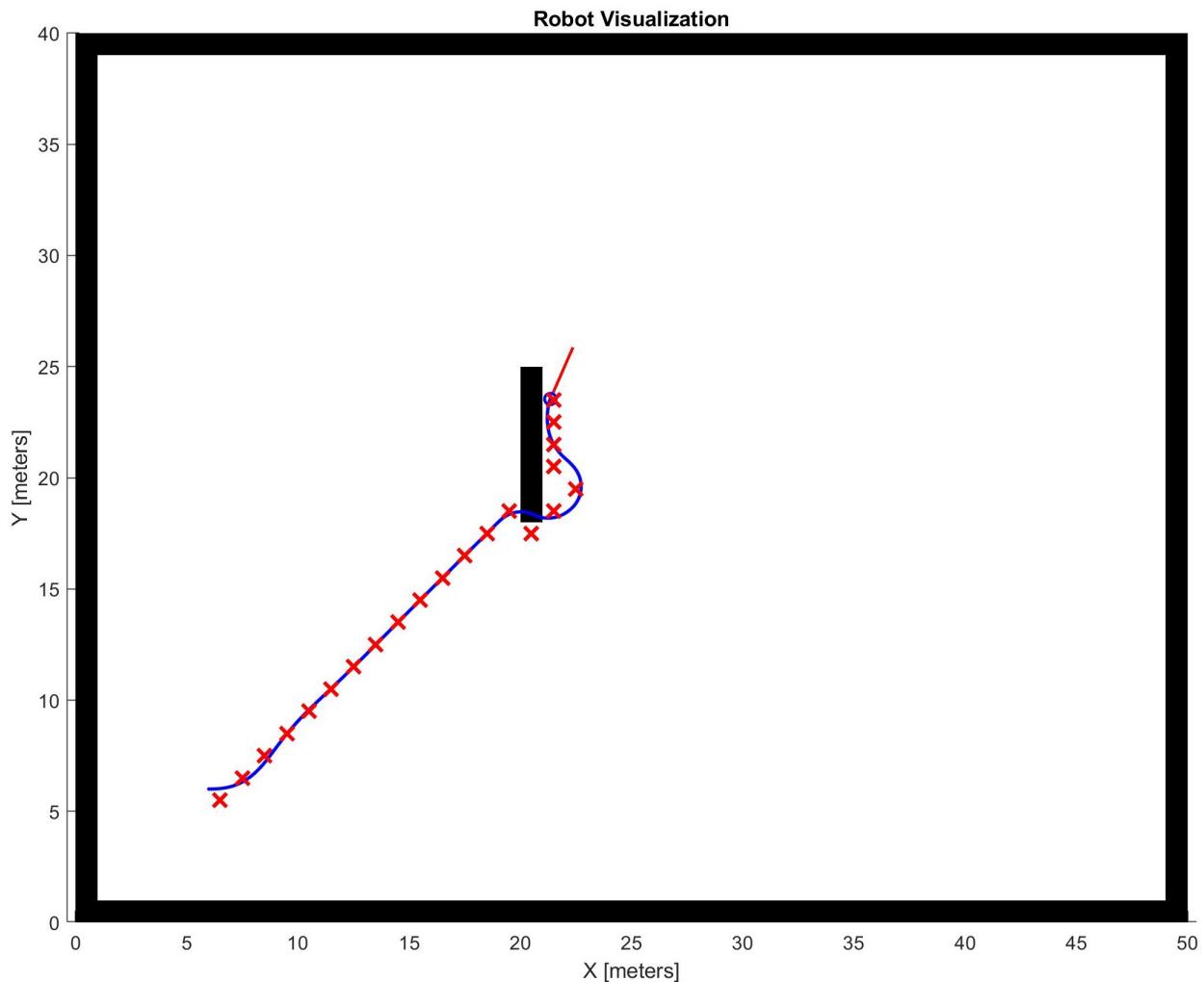
عقب بیاندازد تا مهره‌های بیشتری را بگیرد و سپس اقدام به بردن بازی نماید و در این بین توسط حریف شکست بخورد هر چند انتخاب این مدل تابع هدف باعث دیدن نتایج اولیه بهتری از سوی این اشتباهات امکان دارد در مسیریابی برای ربات هم اتفاق بیفت. به عنوان مثال اگر طبق فاصله $R = -\frac{\|(y-goal)\|_2}{1000}$ ، مسیریابی بهین نخواهد بود و در این مثال همانطور که در شکل ۱۳-۴ مشخص است، تحت تغییر مسیر ناگهانی ربات با مانع اثابت کرده است. دلیل این انتخاب این بوده که حالت $[19]^{T^T}$ جریمه کمتری داشته و ربات علاقه‌مند به عبور از آن بوده و همچنین روی مسافت پیموده شده جریمه‌ای تعیین نشده است. هر چند سریع‌تر و با ۷۷ گام به پاسخ رسیده است. دلیل تقسیم بر ۱۰۰۰ نیز ایجاد نسبت معقول بین پاداش رسیدن و جریمه بوده است.



شکل ۱۳-۴: نتیجه الگوریتم Q-learning برای یادگیری با

حتی اگر بیاییم و تابع پاداشی تعریف کنیم که بسیار نزدیک به تابع پاداش بهین ۱۴-۴ باشد، مجدداً شکست خواهیم خورد. در این قسمت اگر پاداش را برابر مقدار فاصله‌ای بگیریم که به نقطه هدف نزدیک

شده‌ایم باز ربات به پیچ سخت همانند حالت قبل تصادف خواهد کرد.



شکل ۱۴-۴: نتیجه الگوریتم Q-learning برای یادگیری با

در نتیجه این مثال‌ها نشان‌دهنده اهمیت انتخاب صحیح تابع پاداش هستند.

۵-۴-۴ حل مثال با الگوریتم Q-learning

در این قسمت، قصد بر این است که مثالی با ابعاد کوچک با الگوریتم Q-learning حل گردد و مقادیر یاد گرفته شده، تحلیل گردد. ابعاد نقشه ۱۰ در ۱۰ می‌باشد و مکان‌هایی که مانع حضور دارد با ۱ در ماتریس loc نمایش داده شده است. حالت شروع در موقعیت $[2\ 2]^T$ و حالت هدف در موقعیت $[8\ 8]^T$ قرار دارد.

همانطور که مشاهده می‌شود الگوریتم، پس از مدت زمان تقریبی ۳۷ میلی ثانیه، در گام ۱۷۱ متوقف شده است.

loc =

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	1	1	1	1	0
0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

شکل ۱۵-۴: نقشه ورودی داده شده به الگوریتم Q-learning

```

iter 42: delta = 1.364944e-02
iter 43: delta = 1.152361e-02
iter 44: delta = 9.728872e-03
iter 45: delta = 8.213651e-03
iter 46: delta = 6.934418e-03
iter 47: delta = 5.854419e-03
iter 48: delta = 4.942624e-03
iter 49: delta = 4.172836e-03
iter 50: delta = 3.522939e-03
iter 51: delta = 2.974260e-03
iter 52: delta = 2.511034e-03
iter 53: delta = 2.119954e-03
iter 54: delta = 1.789782e-03
iter 55: delta = 1.511033e-03
iter 56: delta = 1.275697e-03
iter 57: delta = 1.077014e-03
iter 58: delta = 9.092749e-04
iter 59: delta = 7.676601e-04
iter 60: delta = 6.481011e-04
iter 61: delta = 5.471627e-04
iter 62: delta = 4.619450e-04
iter 63: delta = 3.899995e-04
iter 64: delta = 3.292591e-04
iter 65: delta = 2.779787e-04
iter 66: delta = 2.346850e-04
iter 67: delta = 1.981340e-04
iter 68: delta = 1.672757e-04
iter 69: delta = 1.412234e-04
iter 70: delta = 1.192286e-04
iter 71: delta = 1.006594e-04
Elapsed time is 0.037116 seconds.
```

شکل ۱۶-۴: زمان طول کشیده برای یافتن جواب توسط Q-learning و مقدار ماکریم تغییر در هر گام

مقدار ماتریس Q که به ابعاد تمام حالت‌ها، ۱۰۰، در تعداد اقدامات، ۸، می‌باشد، در شکل ۱۷-۴ آمده است.

$Q =$

8.9700	-10.0000	7.3749	-10.0000	-10.0000	8.8913	-10.0000	-10.0000
9.3055	-10.0000	-10.0000	7.0730	8.5558	-10.0000	-10.0000	-10.0000
-10.0000	-10.0000	14.1767	7.3749	8.8914	16.4489	-10.0000	-10.0000
16.8631	-10.0000	14.2839	-10.0000	-10.0000	16.0762	-10.0000	-10.0000
16.4904	-10.0000	16.9822	14.1768	16.4489	16.9408	-10.0000	-10.0000
17.3550	-10.0000	17.3550	14.2840	16.0762	19.9803	-10.0000	-10.0000
20.3945	-10.0000	17.6905	16.9823	16.9408	20.3531	-10.0000	-10.0000
20.7673	-10.0000	17.6905	17.3551	19.9804	20.3531	-10.0000	-10.0000
20.7673	-10.0000	17.6906	17.6906	20.3531	20.3531	-10.0000	-10.0000
20.7674	-10.0000	-10.0000	17.6906	20.3531	-10.0000	-10.0000	-10.0000
11.0779	7.0730	9.3055	-10.0000	-10.0000	11.0364	-10.0000	6.9607
11.4506	7.3749	-10.0000	8.9701	10.6637	-10.0000	6.6588	-10.0000
-10.0000	-10.0000	16.8631	9.3056	11.0365	19.4338	6.9607	13.7625
19.8480	14.1768	16.4904	-10.0000	-10.0000	-10.0000	-10.0000	13.8697
-10.0000	14.2840	17.3550	16.8632	19.4338	-10.0000	13.7626	16.5680
-10.0000	16.9823	20.3945	16.4904	-10.0000	-10.0000	13.8698	16.9408
-10.0000	17.3551	20.7673	17.3551	-10.0000	23.7717	16.5681	17.2763
24.1860	17.6906	20.7673	20.3946	-10.0000	23.7717	16.9408	17.2763
24.1860	17.6906	20.7674	20.7674	23.7718	23.7718	17.2764	17.2764
24.1860	17.6906	-10.0000	20.7674	23.7718	-10.0000	17.2764	-10.0000
11.4506	8.9701	11.4506	-10.0000	-10.0000	13.4199	-10.0000	8.8913
13.8341	9.3056	-10.0000	11.0779	11.0365	-10.0000	8.5558	-10.0000
-10.0000	-10.0000	19.8480	11.4507	13.4200	19.8065	8.8914	16.4489
20.2208	16.8632	-10.0000	-10.0000	-10.0000	23.1645	-10.0000	16.0762
23.5787	16.4904	-10.0000	19.8480	19.8066	26.8955	16.4489	16.9408
27.3097	17.3551	-10.0000	-10.0000	23.1645	27.2683	16.0762	19.9803
27.6825	20.3946	24.1860	-10.0000	26.8955	27.5702	16.9408	20.3531
27.9845	20.7674	24.1860	-10.0000	27.2683	27.5702	19.9804	20.3531
27.9845	20.7674	24.1860	24.1860	27.5703	27.5703	20.3531	20.3531

شکل ۱۷-۴: مقدار ماتریس Q پس از پایان الگوریتم Q-learning

ماتریس V مقدار نسبت داده شده به هر حالت را نشان می‌دهد. همانطور که دیده می‌شود حالت‌های مجاور حالت هدف دارای بیشترین مقدار در ماتریس V هستند.

$V =$

8.9700	11.0779	13.4199	13.8341	15.7741	16.1884	16.5611	16.8967	17.1986	17.4704
9.3055	11.4506	13.8341	16.4824	16.8967	19.0982	19.5125	19.8852	20.2208	20.5227
16.4489	19.4338	19.8480	20.2208	19.8852	19.5125	22.7917	23.2059	23.5787	23.9142
16.8631	19.8480	23.1645	23.5787	23.2059	22.7917	23.2059	26.8955	27.3097	27.6825
16.9822	19.4338	26.8955	27.3097	26.8955	23.2059	23.5787	27.3097	31.4553	31.8695
19.9803	20.3945	27.3097	31.4553	36.1075	36.5217	36.8945	36.8945	36.8945	36.5217
20.3945	23.7717	27.6825	31.8695	36.5217	41.6908	42.1050	42.1050	42.1050	41.6908
20.7673	24.1860	27.9845	32.2050	36.8945	42.1050	47.8945	47.8945	47.8945	42.1051
20.7673	24.1860	27.9845	32.2050	36.8945	42.1050	47.8945	42.1051	47.8946	42.1051
20.7674	24.1860	27.9845	32.2050	36.8945	42.1051	47.8946	47.8946	47.8946	42.1051

شکل ۱۸-۴: مقدار ماتریس V پس از پایان الگوریتم Q-learning

همانطور که دیده می‌شود، حالت‌ها به طور نسبتاً بهین همگی سیاستی به سوی نقطه هدف ارائه می‌دهند. هر عدد نماینده المان با همان اندیس در مجموعه اقدامات ۸-۴ می‌باشد.

$P_i =$

1	1	6	3	6	3	3	3	3	3
1	1	1	6	3	6	3	3	3	3
6	6	3	3	3	1	6	3	3	3
1	1	6	3	3	8	1	6	3	3
3	5	6	3	8	2	1	1	6	3
6	3	1	6	6	3	3	3	3	3
1	6	1	1	1	6	3	3	3	8
1	1	1	1	1	1	6	3	8	2
1	1	1	1	1	1	1	1	2	2
1	1	1	1	1	1	5	4	7	2

شکل ۱۹-۴: سیاست نهایی برای هر حالت در پایان الگوریتم Q-learning

مسیر پیشنهادی برای رسیدن از نقطه شروع به هدف، در شکل ۲۰-۴ آمده است.

$path =$

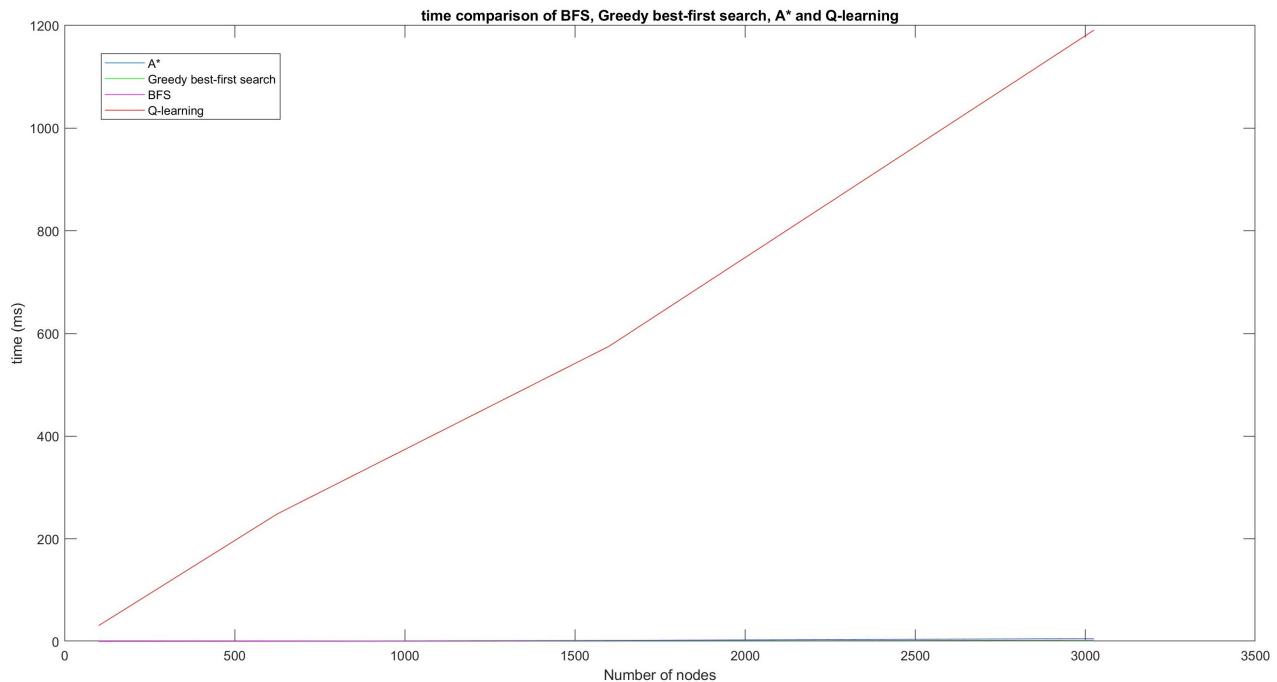
2	2
2	3
2	4
3	5
4	5
5	5
6	4
7	5
7	6
8	7
9	8

شکل ۲۰-۴: مسیر پیشنهادی توسط الگوریتم Q-learning

۶-۴-۴ مقایسه زمانی الگوریتم Q-learning با الگوریتم‌های ابتکاری

حال از نظر زمانی، الگوریتم Q-learning با الگوریتم‌های ابتکاری مقایسه می‌گردد. همانطور که در شکل ۲۱-۴ دیده می‌شود الگوریتم Q-learning زمان بسیار بیشتری نسبت به الگوریتم‌های ابتکاری نیازمند است. در نتیجه برای مسیریابی آنلاین به مشکل خواهیم خورد. برای رفع این مشکل، از مزیت بزرگ الگوریتم Q-learning که در صورت جایی موانع، با به روزرسانیتابع V با تعداد گام بسیار کمی سیاست خود را با محیط وفق می‌دهد، استفاده می‌شود. در نتیجه با یک زمان نسبتاً کم اولیه اگر به سیاست خوبی برسیم، در ادامه کار می‌توان به صورت بلادرنگ، با آپدیت کردن مقادیر الگوریتم در بازه زمانی کم، اصلاح مسیر را انجام داد. همچنین نباید از این موضوع غافل شویم که الگوریتم Q-learning

برای تمام حالت‌ها سیاست بهین را فرا می‌گیرد، در نتیجه اگه ربات با حالتی برود که انتظار نداشتیم، باز مشکلی نخواهد بود.



شکل ۲۱-۴: مقایسه زمانی الگوریتم‌های ابتکاری Q-learning با الگوریتم‌های A*، Greedy best-first search و BFS

فصل پنجم

مسیریابی برای اجماع ربات‌ها

۱-۵ مقدمه

در این قسمت به پیاده‌سازی الگوریتم‌های فصل ۴ برای اجماع ربات‌ها به صورت آنلاین پرداخته می‌شود. منظور از آنلاین بودن برنامه‌ریزی این است که در هر گام زمانی، ربات مجدداً الگوریتم مسیریابی را اجرا نماید. برای مسیریابی با میدان پتانسیل، در ابتدا اجماع با شکل‌دهی ثابت برای عبور از موانع تلاش می‌کند اما پس از اشاره به ایرادات این روش، در ادامه کار شکل‌گیری پلتون شکسته و دوباره به شکل‌گیری تلاش خواهد کرد. همچنین موانع به صورت ثابت و متحرک خواهند بود.

۲-۵ میدان پتانسیل

در این قسمت تمامی مسیریابی‌ها تنها با میدان پتانسیل انجام شده است. در ابتدا شکل‌گیری ثابت و سپس شکل‌گیری با قابلیت شکسته شدن مورد بررسی قرار خواهد گرفت. قابل ذکر است که در هر دو روش تا جای ممکن محاسبات به صورت نامتمرکز^۱ بوده است.

۱-۲-۵ جلوگیری از برخورد با موانع با شکل‌گیری ثابت

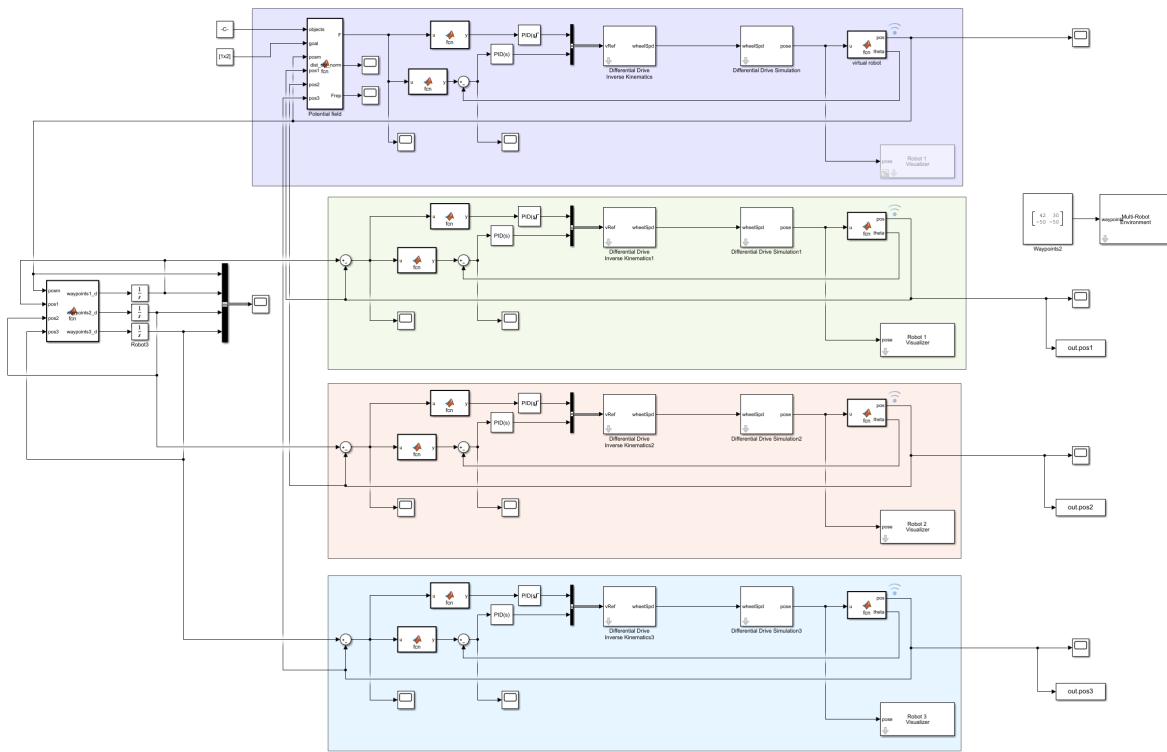
در این قسمت یک شکل‌گیری ثابت برای دسته ربات‌ها در نظر گرفته شده است. ایده اصلی این است که نیروی جاذبه برای رهبر که در مرکز دسته پلتون حضور دارد و نیروی دافعه برای هر ربات محاسبه گردد و در نهایت نیروی کلی به دست آید. در این راستا برای محاسبه دو ایده برای پیاده‌سازی به وجود می‌آید اول اینکه همانند شکل ۱-۵ به صورت متمرکز تمام داده‌ها به محاسبه‌گری که مسئول محاسبه مسیر ربات رهبر است فرستاده شود و نیروهای دافعه و جاذبه در آن محاسبه گردند.

مکان ربات‌ها برای روش متمرکز به رنگ‌های سبز، آبی و قرمز و موانع با رنگ سیاه در شکل ۲-۵ به نمایش در آمده است.

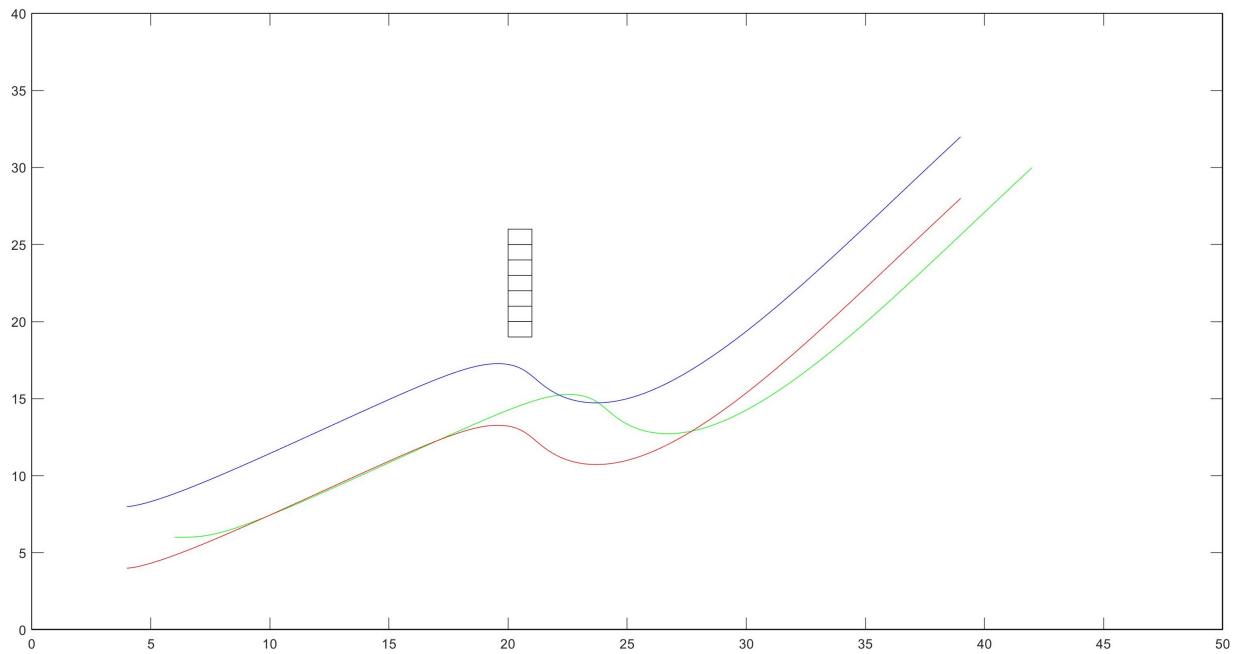
در روش دوم، برای اینکه دوره محاسبات کاهش یابد، همانند شکل ۳-۵ محاسبه نیروی دافعه ناشی از هر ربات، به صورت نامتمرکز و جداگانه در هر ربات محاسبه می‌شود و سپس به ربات رهبر فرستاده می‌شود. محاسبه‌گر ربات رهبر تنها مقادیر را با هم جمع می‌نماید تا به نیروی دافعه برسد. اما نیروی جاذبه همچنان توسط رهبر محاسبه می‌گردد.

مکان ربات‌ها برای روش نامتمرکز در شکل ۴-۵ به نمایش در آمده است.

¹distributed

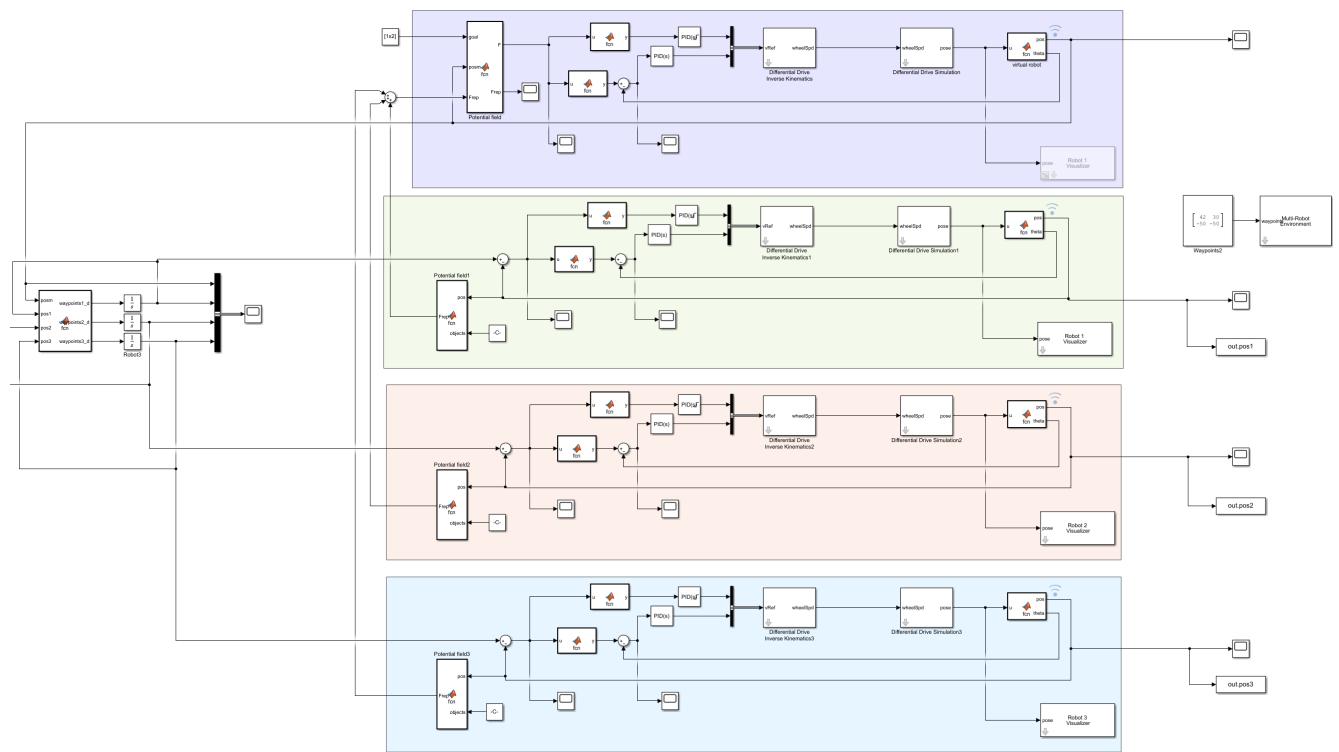


شکل ۱-۵: بلوک‌های شبیه‌سازی در روش شکل‌گیری ثابت با محاسبات متتمرکز

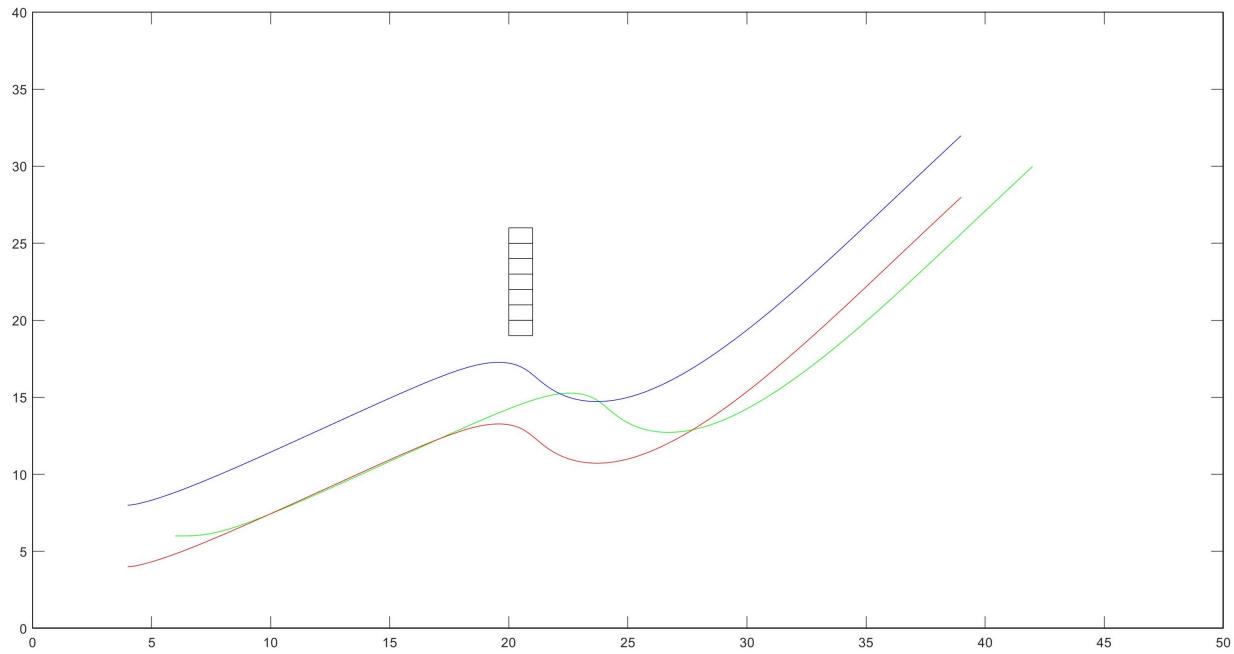


شکل ۲-۵: مکان ربات‌ها در روش شکل‌گیری ثابت با محاسبات متتمرکز

مشکل اصلی این روش این است که امکان دارد هر ربات از جهت مختلفی به طور همزمان به مانع نزدیک شوند، مخصوصاً در شرایطی که مانع متحرک باشد، لذا نیاز داریم هر ربات به تنها یک میدان



شکل ۵-۳: بلوک های شبیه سازی در روش شکل گیری ثابت با محاسبات نامتمرکز

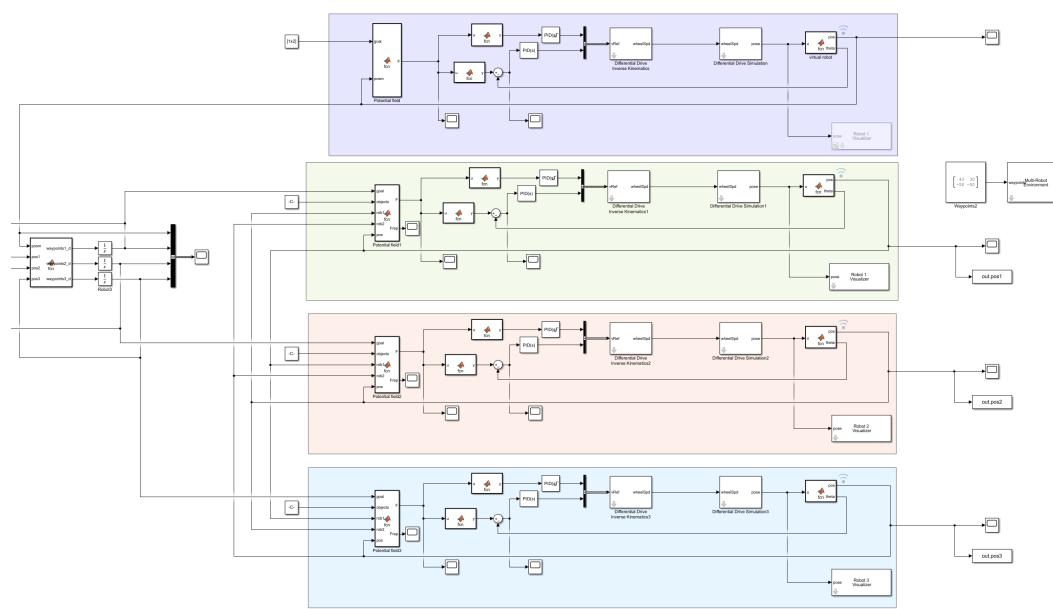


شکل ۴-۵: مکان ربات‌ها در روش شکل‌گیری ثابت با محاسبات نامتمرکز

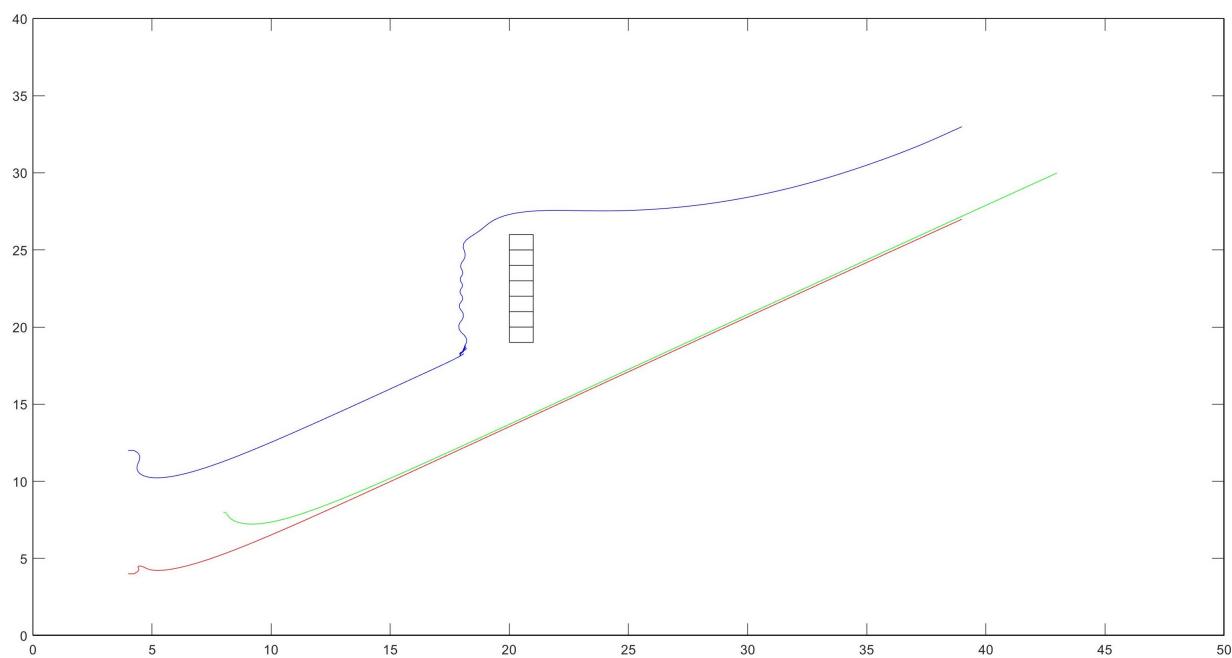
پتانسیل مخصوص به خود را داشته باشد. که در قسمت بعد این ایده پیاده‌سازی گشته است.

۲-۲-۵ جلوگیری از برخورد با موانع با شکل‌گیری شکننده

در این روش هر ربات با میدان پتانسیل مخصوص به خود مسیریابی را انجام می‌دهد و نقطه‌ای که در هر لحظه باید در اجماع قرار گیرد به عنوان نقطه هدف در نظر گرفته می‌شود. بلوک‌های شبیه‌سازی در شکل ۵-۵ و نتیجه حرکت ربات‌ها در شکل ۶-۵ به نمایش در آمده است.



شکل ۵: بلوک‌های شبیه‌سازی در روش شکل‌گیری شکننده



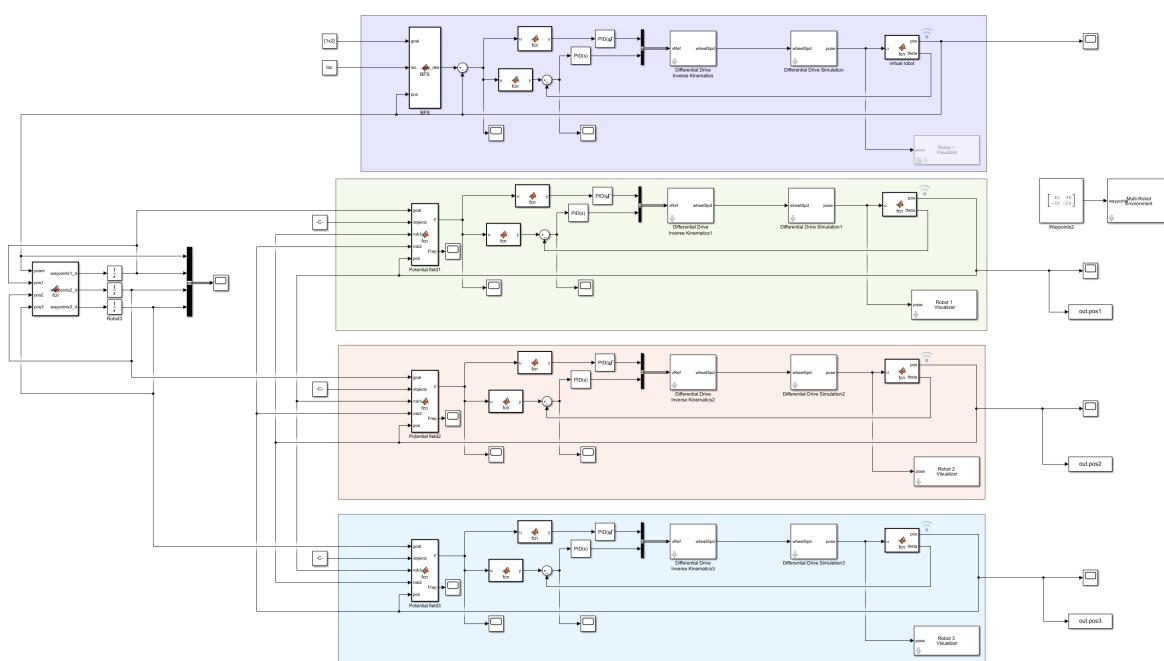
شکل ۶: مکان ربات‌ها در روش شکل‌گیری شکننده

۳-۵ روش‌های ابتکاری

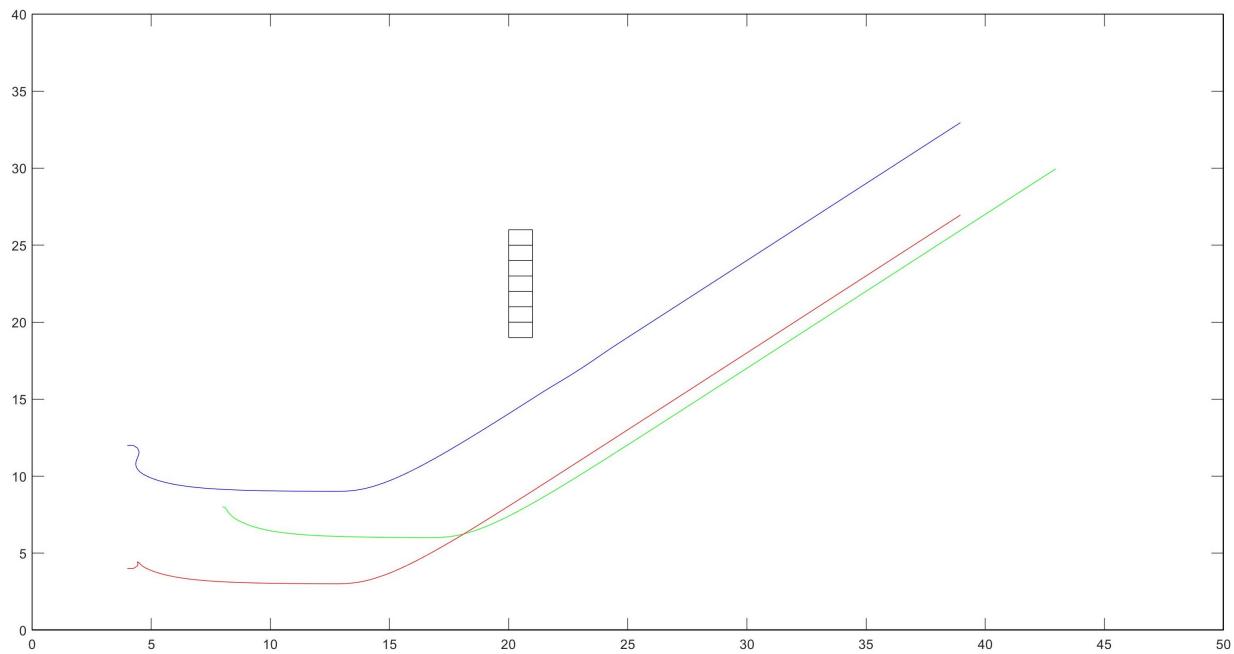
در این قسمت روش‌های ابتکاری فصل ۴ پیاده‌سازی شده‌اند. این مسیریابی‌ها به صورت آنلاین انجام شده است تا از برخورد با موانع متحرک نیز بتوان اجتناب کرد. روش پیاده‌سازی به این صورت است که در عامل رهبر، این مسیریابی انجام و ربات طبق آن حرکت می‌نماید. سایر ربات‌ها همانند بخش ۲-۲-۵ از میدان پتانسیل برای رسیدن به مکان مطلوب تعیین شده از طرف کنترل‌کننده پلتون و دوری از موانع به صورت انفرادی، استفاده می‌کنند.

BFS ۱-۳-۵

نحوه پیاده‌سازی مسیریابی BFS توسط اجمع ربات‌ها در شکل ۷-۵ و مسیر پیموده شده در شکل ۸-۵ قابل مشاهده است.



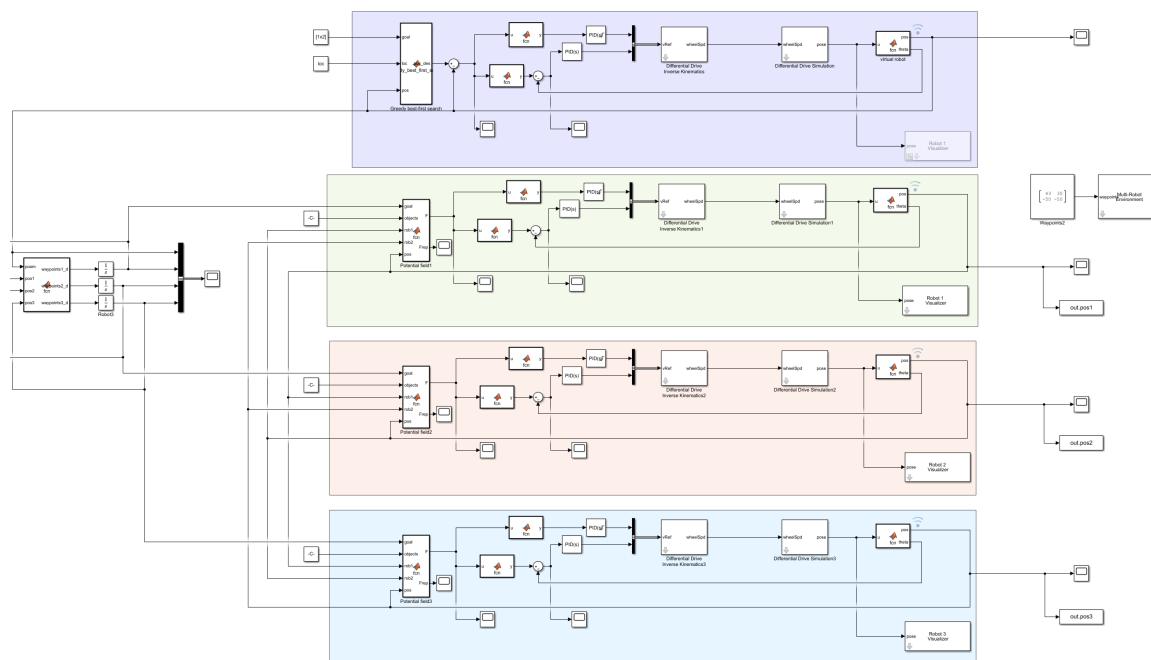
شکل ۷-۵: بلوک‌های شبیه‌سازی در روش BFS



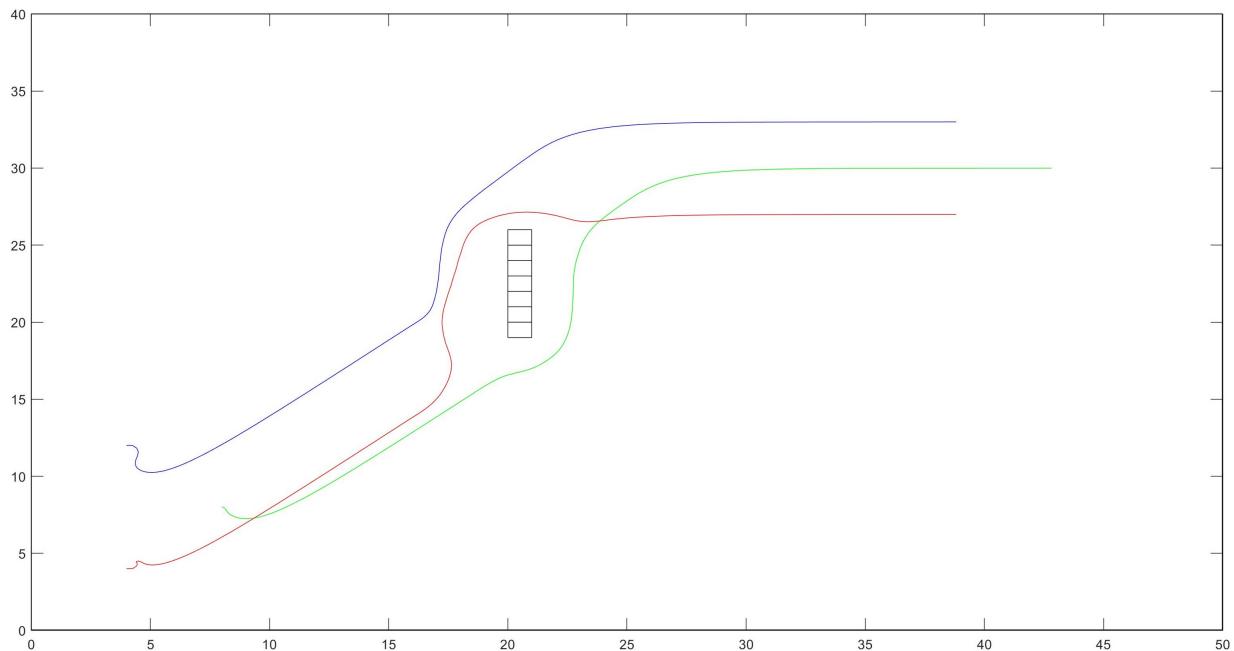
شکل ۸-۵: مکان ربات‌ها در روش BFS

Greedy best-first search ۲-۳-۵

پیاده‌سازی روش Greedy best-first search برای مسیریابی اجماع ربات‌ها همراه با نتیجه آن، به ترتیب در شکل‌های ۹-۵ و ۱۰-۵ آمده است.



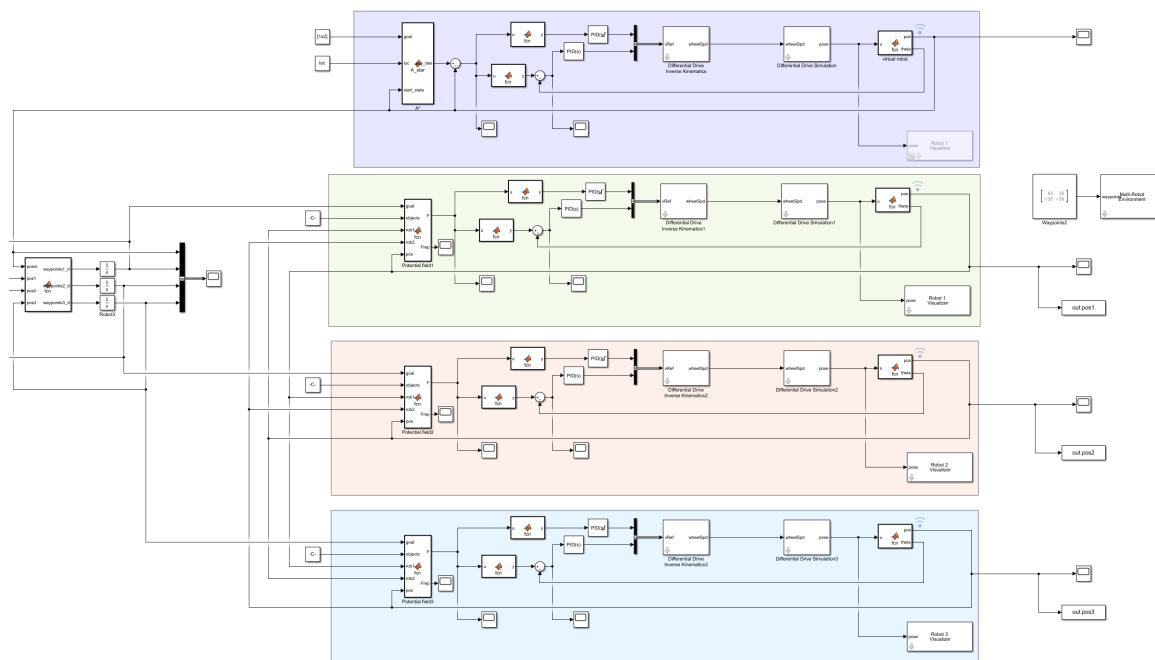
شکل ۹-۵: بلوک‌های شبیه‌سازی در روش Greedy best-first search



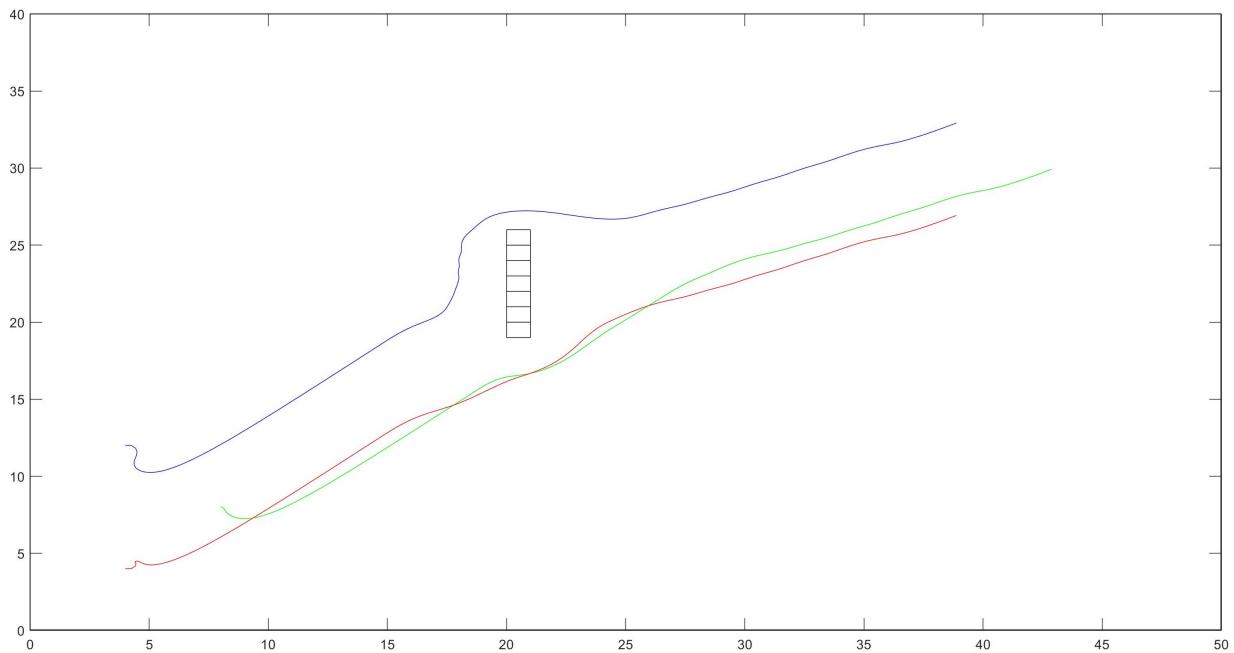
شکل ۱۰-۵: مکان ربات‌ها در روش Greedy best-first search

A^* ۳-۳-۵

روش A^* برای مسیریابی اجماع ربات‌ها در شکل ۱۱-۵ پیاده‌سازی شده است. مسیر پیموده شده توسط هر ربات، در شکل ۱۲-۵ قابل مشاهده می‌باشد.



شکل ۱۱-۵: بلوک‌های شبیه‌سازی در روش A^*



شکل ۱۲-۵: مکان ربات‌ها در روش A^*

Q-learning ۴-۵

در این قسمت نتایج الگوریتم Q-learning بررسی خواهد شد. قابل ذکر است که مقادیر یادگیری به شرح زیر هستند:

$$\cdot / \alpha = \alpha \bullet$$

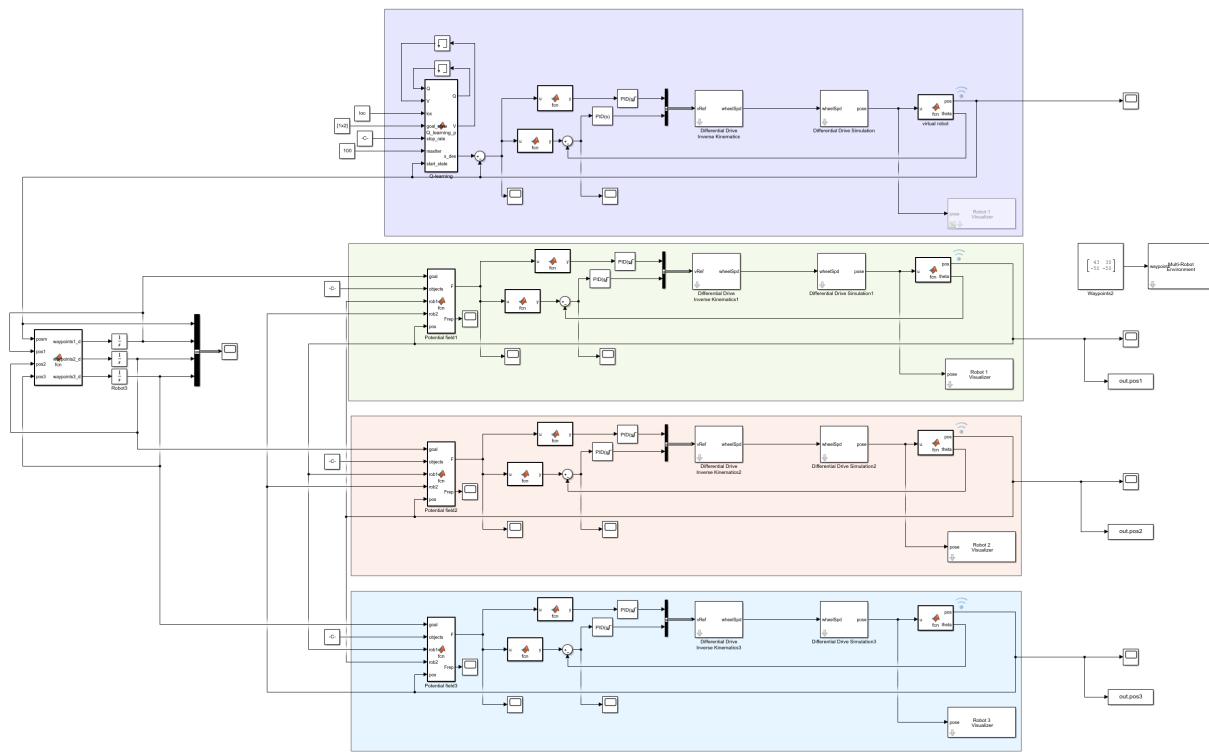
$$\cdot / \gamma = \gamma \bullet$$

$$\bullet \text{ بیشترین تعداد گامها} = 100$$

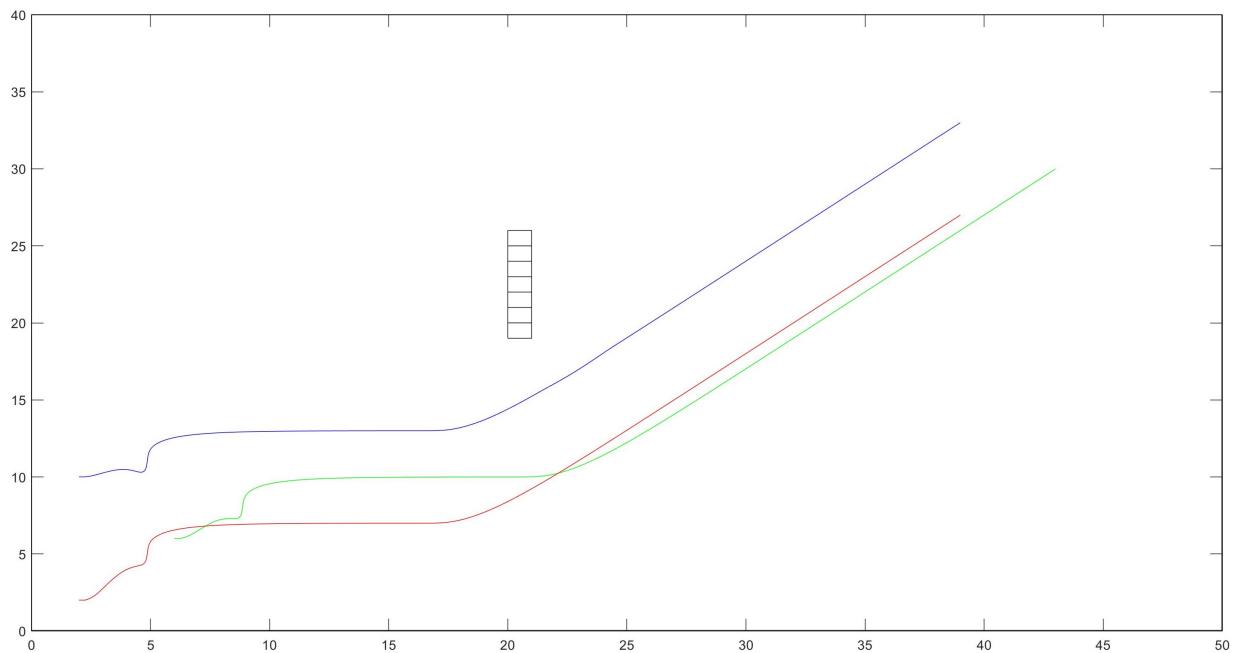
$$\bullet \text{ حد بیشترین اختلاف در تابع } V = 0.0001$$

$$R = -\|a\|_2 \bullet$$

بلوک‌های شبیه‌سازی همانند شکل ۱۳-۵ بسته شده‌اند. همچنین مسیر پیموده شده توسط هر ربات در شکل ۱۴-۵ نشان داده شده است. در پیاده‌سازی آنلاین، مقادیر V و Q نگهداری می‌شوند و در هر دوره محاسباتی تنها به روزرسانی می‌گردند. لذا همانطور که مشخص است در یکی دو سیاست اولیه اخذ شده، هنوز به بهینگی نرسیده‌ایم اما پس از ۲ انتخاب، سیاست‌ها بهینه هستند.



شکل ۱۳-۵: بلوک‌های شبیه‌سازی در روش Q-learning



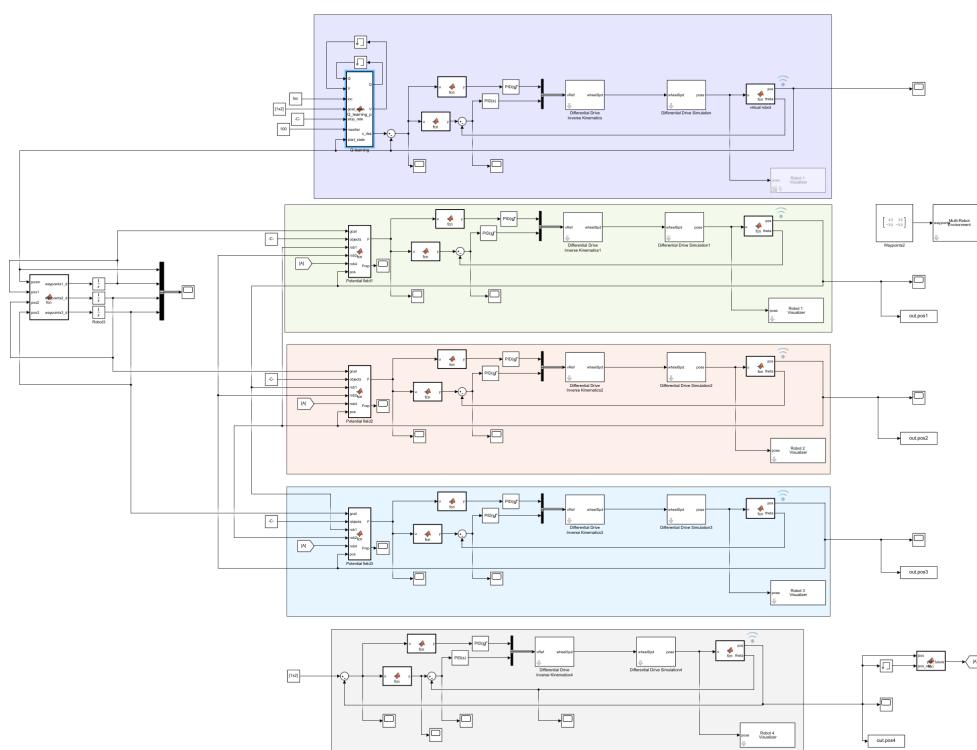
شکل ۱۴-۵: مکان ربات‌ها در روش Q-learning

۵-۵ مسیریابی در حضور موانع متحرک

در ادامه نتایج گرفته شده در برابر موانع متحرک برای الگوریتم‌های بالا به نمایش در آمده‌اند. قابل ذکر است که فیلم پاسخ‌های گرفته شده در [۲] موجود می‌باشد. روش مواجهه به موانع متحرک به این صورت بوده است که جابه‌جایی آن‌ها در یک بازه زمانی محاسبه گردد و سپس طبق جابه‌جایی انجام شده در لحظه قبل و داشتن موقعیت لحظه حال، موقعیت مانع متحرک در لحظه بعدی تقریب زده شود. این روش بسیار ساده است و قابل ارتقا و دقیق‌تر شدن می‌باشد.

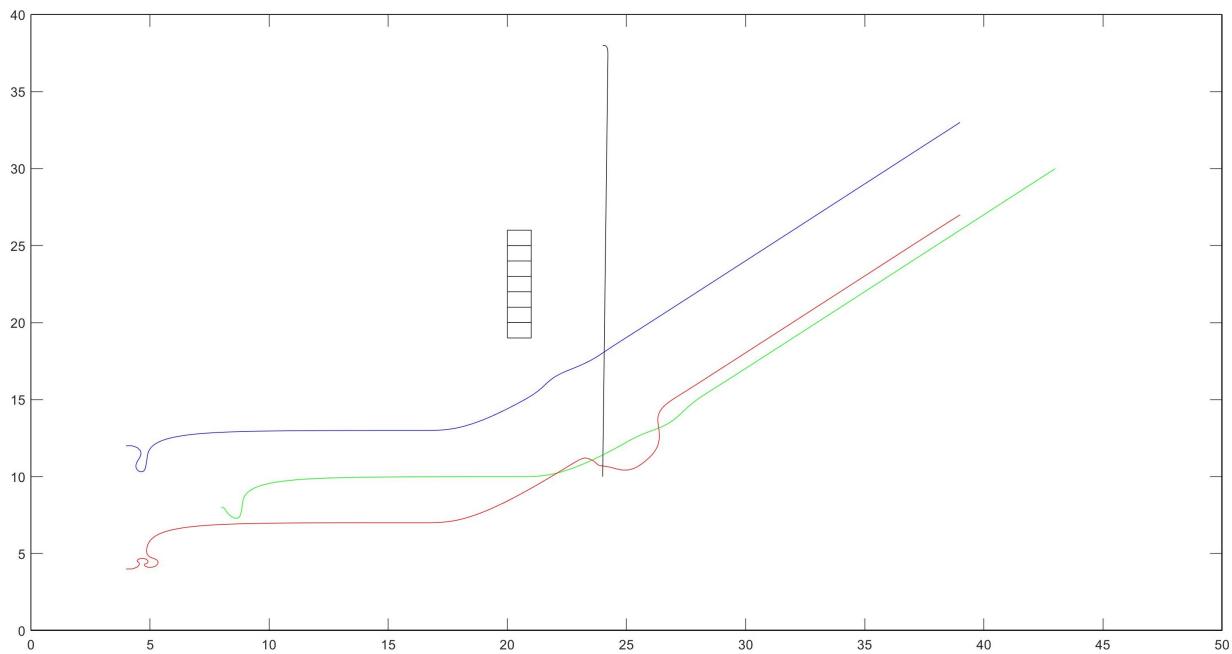
$$X_{n+1} = X_n + (X_n - X_{n-1}) = 2X_n - X_{n-1} \quad (1-5)$$

شبیه‌سازی برای Q-learning در عکس ۱۵-۵ نشان داده شده است.

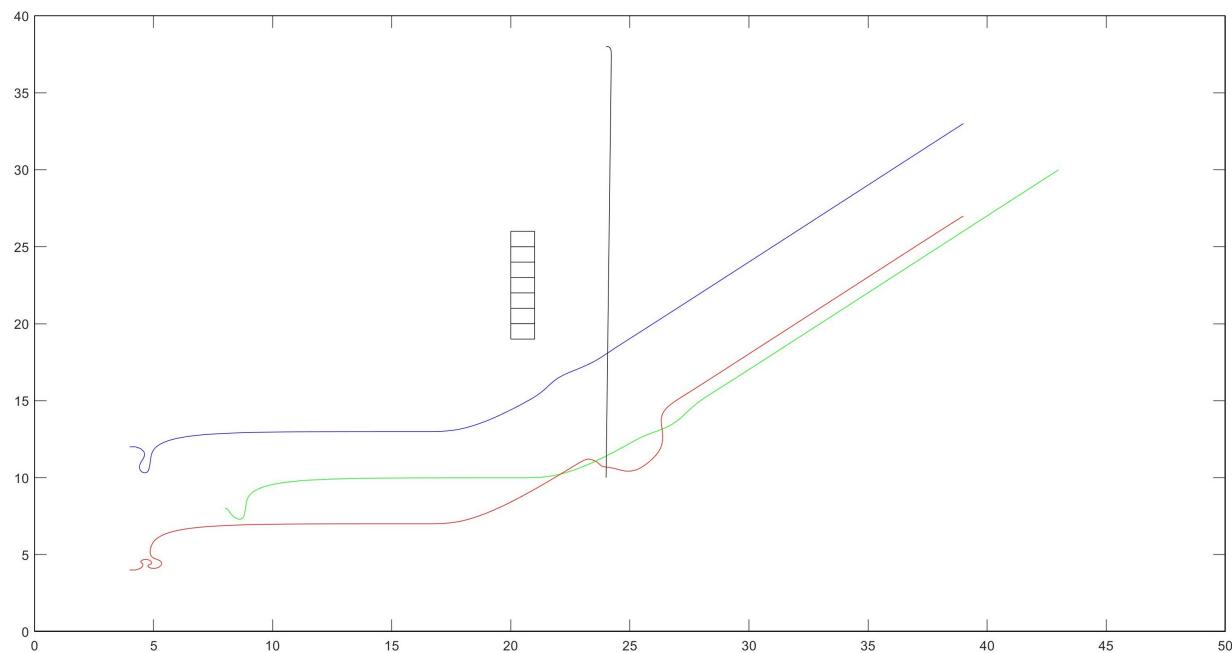


شکل ۱۵-۵: بلوک‌های شبیه‌سازی در روش Q-learning همراه با مانع متحرک

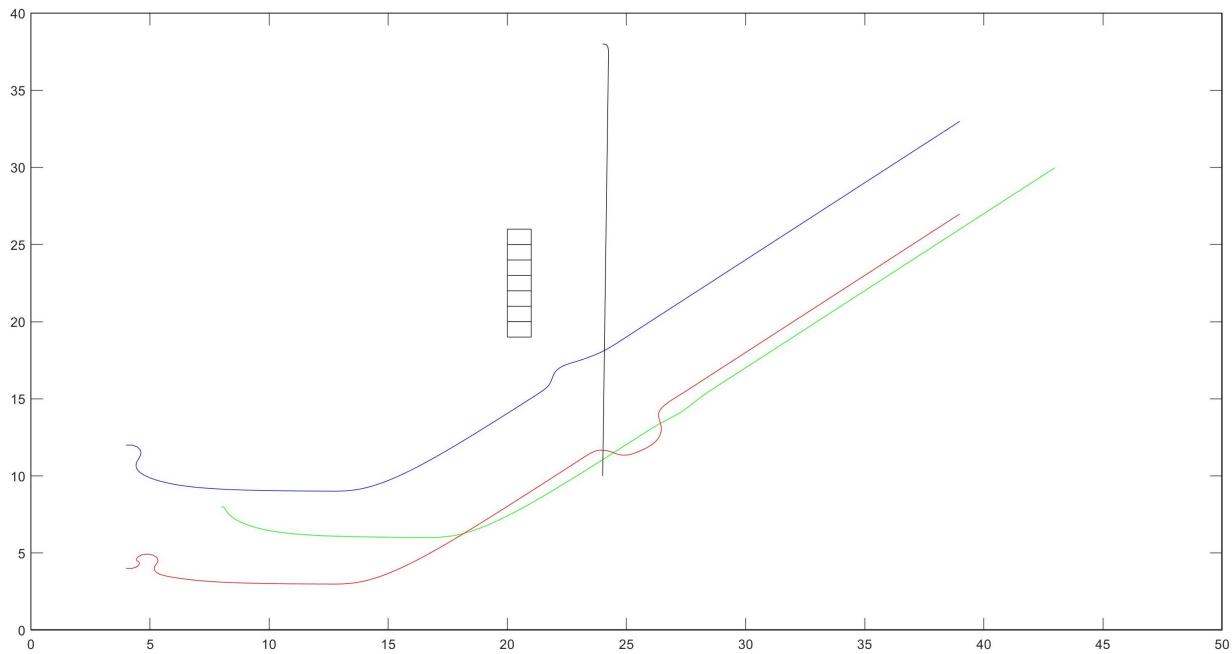
نتایج برای هر الگوریتم نیز در عکس‌های زیر نشان داده شده‌اند. مسیرهای سبز، قرمز و آبی توسط ربات‌های پلتون و مسیر مشکی رنگ توسط مانع متحرک که خود نیز یک ربات است پیموده شده‌اند.



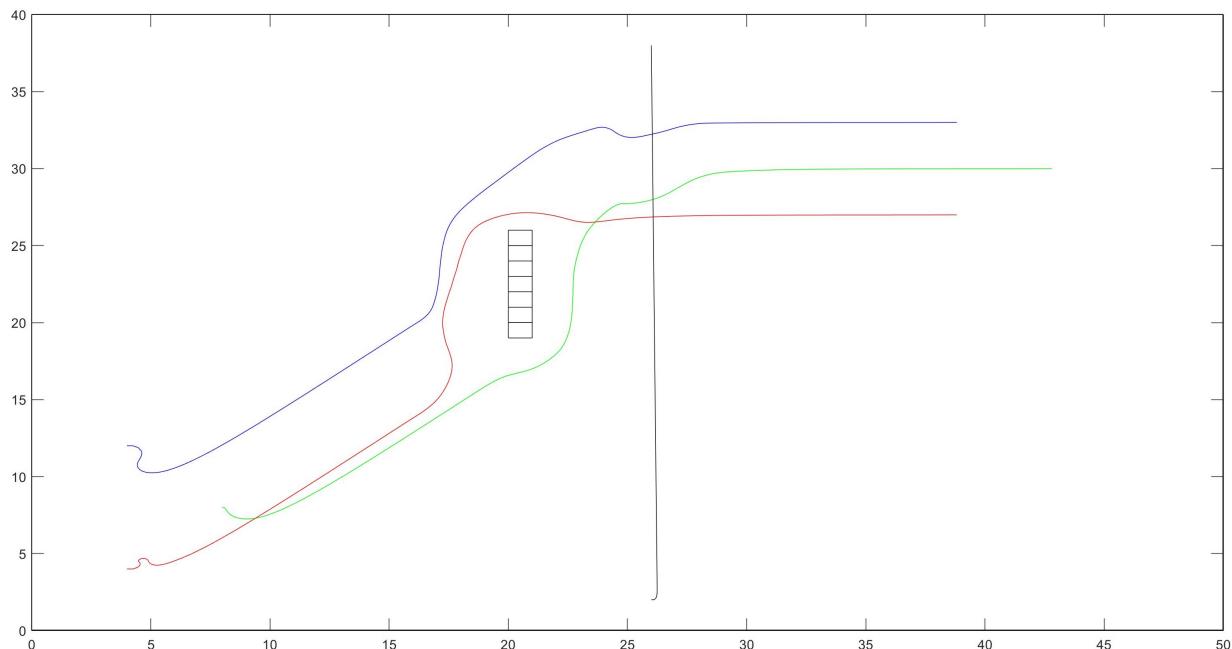
شکل ۱۶-۵: مکان ربات‌ها در روش Q-learning همراه با مانع متحرک



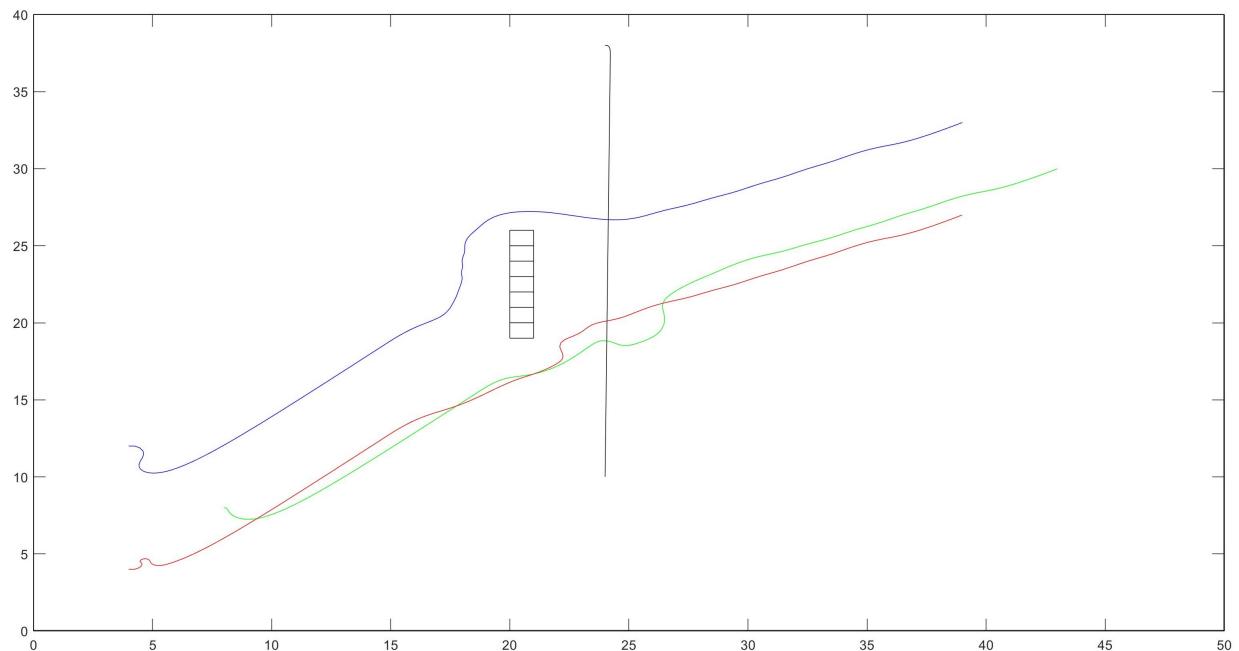
شکل ۱۷-۵: مکان ربات‌ها در روش میدان پتانسیل همراه با مانع متحرک



شکل ۱۸-۵: مکان ربات‌ها در روش BFS همراه با مانع متحرک



شکل ۱۹-۵: مکان ربات‌ها در روش Greedy best-first search همراه با مانع متحرک



شکل ۲۰-۵: مکان ربات‌ها در روش A^* همراه با مانع متحرک

فصل ششم

جمع‌بندی و نتیجه‌گیری و پیشنهادات

۱-۶ نتیجه‌گیری

این پایان‌نامه موفق به انجام مسیریابی و کنترل اجماع به طور مستقل و در نهایت ادغام این دو با یکدیگر بوده است. همچنین از مباحث گسترده‌ای همچون کنترل دیجیتال، کنترل مدرن، رباتیک پیشرفته، جبر خطی، نظریه گراف، طراحی الگوریتم، هوش مصنوعی و یادگیری تقویتی در این پروژه بهره برده شده است. همچنین علاوه بر نتایج موفق به خصوص در حوزه Q-learning و الگوریتم‌های ابتکاری، در بحث تئوری نیز پایداری الگوریتم مطرح شده اثبات گردیده است. در ادامه جا دارد که نکات مهم این پایان‌نامه مرور گردند:

۱. چالش محاسبه سینماتیک ربات که با استفاده از مدل ساده‌تر به دست آمد.
۲. چالش برطرف کردن کنترل زاویه هنگام ناپیوستگی بین π و $-\pi$ – که با پیدا کردن کمترین اختلاف در هر دو جهت عقربه‌های ساعت و خلاف عقربه‌های ساعت به دست آمد.
۳. در روش پیشنهادی برای کنترل اجماع در این پایان‌نامه دیگر نیازی به اینکه همانند پایان‌نامه‌های گذشته، مجموع درایه‌های روی ماتریس شکل‌دهی برابر با صفر باشند نبود و همچنین ربات‌ها دقیقاً به مکان خواسته شده می‌رفتند در حالی که در روش پیشنهادی گذشته، قرارگیری ربات‌ها علاوه بر ماتریس F به ماتریس L نیز وابسته بود. همچنین پایداری این روش اثبات گردید و با یک حقه قرارگیری در مختصات $y-x$ را نیز نسبت به هم مستقل کردیم تا راحت‌تر بتوان ماتریس F مورد نظر را تشکیل داد.
۴. مشکل تاخیر پاسخ توسط عامل‌ها نسبت به عامل رهبر که با مجازی در نظر گرفتن رهبر حل گردید.
۵. کد بسیار مرتب پیاده‌سازی شده به طوری که با ساختن نقشه‌های جدید تنها با وارد کردن مختصات موانع یا تغییر در ابعاد نقشه، فایل‌ها قابل اجرا باشند.
۶. یکی از چالش‌های مهم، نحوه تعریف محیط پیوسته، شامل حالات و اقدامات ممکن، برای الگوریتم‌های مسیریابی بود.
۷. پیاده‌سازی الگوریتم‌های ابتکاری و Q-learning و مقایسه آن‌ها یکی از اهداف مهم پایان‌نامه بود.

۸. تنظیم پارامترهای الگوریتم Q-learning و به خصوص تابع پاداش آن بسیار پیچیده و دشوار بود که با بررسی حالت‌های مختلف و به خصوص اشتباهات رایج، توانستیم با زمان مناسب به پاسخ اغلب بهین دست یابیم.

۹. در نهایت نیز، پیاده‌سازی الگوریتم‌های مسیریابی بر روی دسته ربات‌ها در حضور موانع ثابت و متحرک، و همچنین تخمین مکان ربات‌های خارج دسته، نتیجه نهایی این پایان‌نامه بود.

۲-۶ پیشنهادات

در آینده می‌توان در این زمینه‌ها فعالیت نمود:

- در ابتدا می‌توان سعی بر کنترل دینامیکی ربات‌ها و یا خودرو به صورت تکی نمود. همچنین از کنترل پیش‌بین مدل^۱ برای کنترل تک ربات یا حتی اجماع می‌توان بهره برد.
- در الگوریتم اجماع از روش تک انتگرال گیر استفاده شده است. می‌توان از روش‌های دو انتگرال گیری برای کاهش تاخیر پاسخ بقیه ربات‌ها نسبت به ربات رهبر بهره برد. همچنین روش‌هایی همانند تخمین موقعیت ربات رهبر، و سعی به رسیدن به موقعیت آن در چند گام بعدتر نیز تلاش نمود.
- در خصوص توضیحات داده شده در مورد گراف‌ها می‌توان الگوریتم‌های مناسبی برای تولید گراف اجماع به طوری که نسبت به قطع ارتباط مقاوم باشند دست یافت.
- می‌توان الگوریتم‌های مسیریابی دیگری همچون RRT^* و RRT را پیاده‌سازی نمود.
- در تشکیل محیط و حالات، به جای انتخاب مختصات با اعداد صحیح، به صورت تصادفی نقطه‌دهی کرد و وجود مانع یا نبود آن را بین حالت‌ها بررسی کرد. سپس پاسخ الگوریتم‌های A^* و BFS و سایر الگوریتم‌ها را بررسی نمود.
- در زمینه یادگیری تقویتی، می‌توان از روش‌های دیگر نیز استفاده نمود و نتایج را بررسی کرد تا به بهترین روش دست یابیم.
- تابع پادash‌های متفاوت دیگر را نیز در الگوریتم Q-learning بررسی کرد و حتی اهدافی دیگر را نیز علاوه بر کوتاهی مسیر به تابع پاداش اضافه نمود. به عنوان مثال می‌توان به تلاش برای کم

¹model predictive control

کردن تغییر جهت در ربات با جرمیه کردن آن به ازای اندازه تغییر در $\theta_{desired}$ یا کاهش تغییرات شتاب وارد^۲ به ربات در مسیر برای راحتی سرنوشتینان، اشاره نمود.

- در آخر نیز می‌توان یادگیری تقویتی را به جای طرح مسیر، بر روی ولتاژ چرخ‌های ربات، اعمال نمود و در نتیجه هوشمندی ربات را افزایش داد. هر چند باید در کنار آن راهی برای امنیت خودرو نسبت به تصادفات نیز پیدا کرد، زیرا در حال حاضر اثبات پایداری برای الگوریتم‌های تقویتی به خصوص در سطح کنترل چرخ‌ها موجود نمی‌باشد.

²jerk

مراجع و مراجع

- [1] Baskar, Lakshmi Dhevi, De Schutter, Bart, Hellendoorn, J, and Papp, Zoltan. Traffic control and intelligent vehicle highway systems: a survey. *IET Intelligent Transport Systems*, 5(1):38–52, 2011.
- [2] Bazargani, Roozbeh. Ee-bsc-thesis. <https://github.com/Roozbeh-Bazargani/EE-BSc-Thesis>, 2021.
- [3] Bondy, John Adrian, Murty, Uppaluri Siva Ramachandra, et al. *Graph theory with applications*, vol. 290. Macmillan London, 1976.
- [4] Cormen, Thomas H, Leiserson, Charles E, Rivest, Ronald L, and Stein, Clifford. *Introduction to algorithms*. MIT press, 2009.
- [5] Diestel, Reinhard. Graph theory 3rd ed. *Graduate texts in mathematics*, 173, 2005.
- [6] Jadbabaie, Ali, Lin, Jie, and Morse, A Stephen. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on automatic control*, 48(6):988–1001, 2003.
- [7] Lefkopoulos, Vasileios and Kamgarpour, Maryam. Using uncertainty data in chance-constrained trajectory planning. in *2019 18th European Control Conference (ECC)*, pp. 2264–2269. IEEE, 2019.
- [8] Li, Shengbo Eben, Zheng, Yang, Li, Keqiang, Wang, Le-Yi, and Zhang, Hongwei. Platoon control of connected vehicles from a networked control perspective: Literature

- review, component modeling, and controller synthesis. *IEEE Transactions on Vehicular Technology*, 2017.
- [9] Perez, Javier Andreu, Deligianni, Fani, Ravi, Daniele, and Yang, Guang-Zhong. Artificial intelligence and robotics. *arXiv preprint arXiv:1803.10813*, 147, 2018.
- [10] Russell, Stuart and Norvig, Peter. *Artificial intelligence: a modern approach*. 2002.
- [11] Salem, Farhan A. Dynamic and kinematic models and control for differential drive mobile robots. *International Journal of Current Engineering and Technology*, 3(2):253–263, 2013.
- [12] Sedigh, A Khaki. *Modern control systems*, 2003.
- [13] Spong, Mark W, Hutchinson, Seth, Vidyasagar, Mathukumalli, et al. *Robot modeling and control*. 2006.
- [14] Strang, G. *Linear algebra and its applications* 1harcourt brace jovanovich. *San Diego, Calif*, 19882:98–99, 1988.
- [15] Sutton, Richard S, Barto, Andrew G, et al. *Introduction to reinforcement learning*, vol. 135. MIT press Cambridge, 1998.
- [16] Watkins, Christopher JCH and Dayan, Peter. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.
- [17] West, Douglas Brent et al. *Introduction to graph theory*, vol. 2. Prentice hall Upper Saddle River, 2001.