



Real-Time Tangent Based Path Planning for Autonomous Robots in Convex Obstacle Environments

Hesamedin Moedi^۱, Mahdi Mozaffari^۲, Abolfazl Manafi^۳, Fatemeh Mahdavi G.^۴, Saeed Shamaghdari^{۱*}

^۱ Master's student, Iran University of Science and Technology, Tehran, Iran

^۲ Master's degree, Iran University of Science and Technology, Tehran, Iran

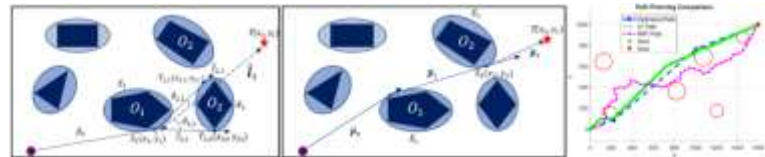
^۳ PhD, Iran University of Science and Technology, Tehran, Iran

^۴ Associate Professor, Iran University of Science and Technology, Tehran, Iran (*Correspondence: shamaghdari@iust.ac.ir)

HIGHLIGHTS

- Significant reduction in computational cost through elliptical obstacle modeling
- Path processing time at least ۶٪ lower than A and ۲٪ lower than RRT
- Capability for real-time execution in large-scale environments
- Potential extension of the method to ۳-D environments

GRAPHICAL ABSTRACT



ARTICLE INFO

Article history:

Article Type: Research paper

Received: ۱۰ October ۲۰۲۰

Received in revised form: ۱۹ December ۲۰۲۰

Accepted: ۰۷ January ۲۰۲۱

Available online: ۲۱ January ۲۰۲۱

*Correspondence:
shamaghdari@iust.ac.ir

Keywords:

Real-Time Path Planning

Applied Geometry

Autonomous Robot

Optimization

ABSTRACT

This study introduces a novel geometric algorithm for online path planning of autonomous robots in two-dimensional environments with convex obstacles. The proposed method models obstacles using optimized inscribed ellipses and generates candidate trajectories through tangent lines. By doing so, the planning problem at each step is confined to a convex region, and the optimal path is determined by minimizing a three-term cost function that incorporates path length, angular deviation, and obstacle clearance. This formulation not only simplifies the computational process but also supports rapid, real-time path updates in dynamic environments.

Extensive simulations confirm that the algorithm considerably outperforms well-established approaches such as A*, RRT*, and ACO. In the first test environment, it achieved an execution time of only ۰.۰۳۲ s (versus ۳.۹۸۱ s for A* and ۴.۱ s for RRT*), a path length of ۱۸۹۶.۹ m (shorter than ۱۹۰۲.۲ m for RRT*), and a smoothness value of ۰.۲۲۰ (compared with ۶۹۶.۶ for A* and ۰.۱۴۷ for RRT*). In the second environment, the computation time was ۰.۱۲۴ s and the smoothness ۰.۴۰, representing at least a ۶٪ improvement over A* and a ۲٪ improvement over RRT*. Furthermore, the method reduced the average energy consumption by ۰-۱۰٪ compared with competing algorithms.

Overall, the findings demonstrate that the proposed algorithm produces shorter, smoother, and more energy-efficient paths while significantly reducing computational cost. These advantages make it a promising candidate for large-scale, real-time robotic applications, with potential for further extension to three-dimensional environments.

Cite this article: Moedi H[©], Mozaffari M[©], Manafi A, Mahdavi F[©], Shamaghdari S[©]. Real-Time Tangent Based Path Planning for Autonomous Robots in Convex Obstacle Environments. Journal of Aerospace Mechanics. ۲۰۲۰; ۲۱ (۴): ۷۷-۸۷. DOI<https://doi.org/10.4۷۱۷۶/MAJ.۲۰۲۰.۱۰۲۰>

© Author(s) retain the copyright and full publishing rights

Publisher: Imam Hossein Comprehensive University.





مسیریابی برخط ربات خودمختار در محیط با موانع محدب مبتنی بر هندسه مماسی

سید حسام‌الدین معدی^۱، مهدی مظفری^۲، ابوالفضل منافی^۳، فاطمه مهدوی گل‌میشه^۴، سعید شمس‌قدری^{۵*}

^۱ دانشجوی ارشد، دانشگاه علم و صنعت ایران تهران، ایران (hesamedin.moedi@gmail.com)

^۲ دانشجوی ارشد، دانشگاه علم و صنعت ایران تهران، ایران (mahdimzaffari@gmail.com)

^۳ کارشناسی ارشد، دانشگاه علم و صنعت ایران تهران، ایران (manafi.abolfazl.۱۳۷۶@gmail.com)

^۴ دکتری، دانشگاه علم و صنعت ایران تهران، ایران (ftmmahdavi@gmail.com)

^۵ دانشیار، دانشگاه علم و صنعت ایران تهران، ایران (نویسنده مسئول: shamaghdari@iust.ac.ir)

برجسته‌ها

- کاهش چشمگیر هزینه محاسباتی با مدل‌سازی بیضوی موانع.
- زمان پردازش مسیر حداقل ۶۰ درصد کمتر از A^* و ۲۰ درصد کمتر از RRT* است.
- قابلیت اجرای برخط در محیط با مقیاس بزرگ.
- امکان گسترش روش به محیط‌های سه‌بعدی.

مشخصات مقاله

تاریخچه مقاله:

نوع مقاله: علمی پژوهشی

دریافت: ۲۳ مهر ۱۴۰۴

بازنگری: ۲۸ آذر ۱۴۰۴

پذیرش: ۱۷ دی ۱۴۰۴

ارائه آنلاین: ۲۳ مهر ۱۴۰۴

*نویسنده مسئول:

shamaghdari@iust.ac.ir

کلید واژه‌ها:

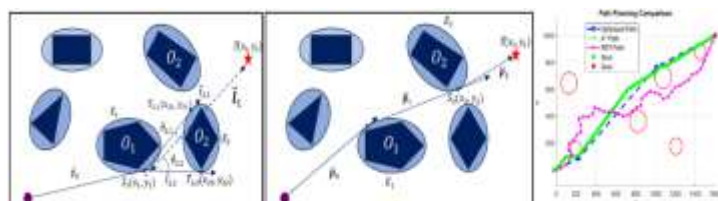
مسیریابی برخط

هندسه کاربردی

ربات خودمختار

بهینه‌سازی

چکیده گرافیکی



چکیده

در این پژوهش یک الگوریتم هندسی نوآورانه برای مسیریابی برخط ربات‌های خودمختار در محیط‌های دوبعدی با موانع محدب ارائه می‌شود. ایده اصلی روش پیشنهادی استفاده از بیضی‌های محاطی بهینه برای مدل‌سازی موانع و به‌کارگیری خطوط مماسی به‌عنوان مسیرهای کاندید است. بدین ترتیب، مسئله مسیریابی در هر گام به یک حوزه‌ی محدب کاهش‌یافته و انتخاب مسیر نهایی از طریق کمینه‌سازی یک تابع هزینه‌ی سه‌جزئی (طول مسیر، میزان انحراف زاویه‌ای و فاصله از موانع) صورت می‌گیرد. این رویکرد نه تنها سادگی محاسبات را تضمین می‌کند؛ بلکه امکان به‌روزرسانی سریع و برخط مسیر در محیط‌های پویا را فراهم می‌سازد. یافته‌های شبیه‌سازی نشان می‌دهد که الگوریتم پیشنهادی به طور چشمگیری بر روش‌های مرسوم A^* ، RRT* و ACO برتری دارد. در محیط نخست، زمان اجرا تنها ۰/۳۲ ثانیه (در برابر ۳/۹۸۱ و ۴/۱ ثانیه)، طول مسیر ۱۸۹۶/۹ کوتاه‌تر از RRT* با ۱۹۵۲/۲ متر و نرمی مسیر ۰/۲۲۵ (در برابر ۶۹۶/۶ و ۵/۱۴۷) ثبت شد. در محیط دوم نیز زمان محاسباتی ۰/۱۲۴ ثانیه و نرمی مسیر ۰/۴ گزارش گردید که حداقل ۶۰٪ سریع‌تر از A^* و حدود ۲۰٪ سریع‌تر از RRT* است. علاوه بر این، مصرف انرژی در مسیرهای الگوریتم پیشنهادی به طور متوسط بین ۵ تا ۱۰ درصد کمتر از سایر مسیرها بود. نتایج نشان می‌دهد الگوریتم پیشنهادی با ترکیب هندسه مماسی و مدل‌سازی بیضوی موانع، قادر است مسیرهایی کوتاه‌تر، نرم‌تر و بهینه‌تر ایجاد کرده و درعین حال زمان پردازش را به طور معناداری کاهش دهد. بدین ترتیب این الگوریتم، گزینه‌ای کارآمد برای کاربردهای مقیاس بالا و برخط محسوب شده، در ضمن قابلیت توسعه به محیط‌های سه‌بعدی را نیز داراست.

استناد: معدی، سید حسام‌الدین^۱، مظفری، مهدی^۲، منافی، ابوالفضل، مهدوی گل‌میشه، فاطمه^۳، شمس‌قدری، سعید^۴. مسیریابی برخط ربات خودمختار در محیط با موانع محدب مبتنی بر هندسه مماسی. مکانیک هوا/فضا. ۱۴۰۴؛ ۲۱ (۳): ۷۷-۸۷.

<https://doi.org/10.47176/MAJ.2020.1020>

© نویسنده(گان) حق نشر و حقوق کامل انتشار را برای خود محفوظ می‌دارند.

ناشر: دانشگاه جامع امام حسین (ع).



۱- مقدمه

برنامه‌ریزی مسیر^۱ یک جنبه اساسی در رباتیک و وسایل نقلیه خودران است که شامل یافتن کوتاه‌ترین و ایمن‌ترین مسیر از مبدأ به مقصد در یک محیط معین است. هدف اصلی یک الگوریتم برنامه‌ریزی مسیر، اجتناب از برخورد با موانع و درعین حال بهینه‌سازی معیارهای عملکردی مانند طول مسیر، زمان اجرا و مصرف انرژی است. با افزایش پیچیدگی محیط‌ها و شرایط متغیر، نیاز به الگوریتم‌های مسیریابی سریع و بهینه در محیط‌های پویا و پیچیده به طور فزاینده‌ای احساس می‌شود. این الگوریتم‌ها باید کارایی و ایمنی حرکت ربات‌ها را در محیط‌های پیچیده و تغییرپذیر بهبود بخشند [۱-۴]. ربات‌های خودمختار در انجام وظایف باید قادر باشند مسیریابی را انتخاب کنند که ضمن اجتناب از برخورد با موانع، زمان و مصرف انرژی را بهینه کنند [۵-۶]. این چالش به‌ویژه در صنایعی مانند لجستیک، حمل‌ونقل و مراقبت‌های بهداشتی که حرکت دقیق و تصمیم‌گیری در زمان واقعی ضروری است، اهمیت دارد. سیستم‌های خودمختار نه تنها باید با محیط‌های ایستا تعامل داشته باشند، بلکه توانایی سازگاری با موانع پویا و شرایط پیش‌بینی نشده را نیز داشته باشند. در این زمینه، الگوریتم‌های کارآمد که بتوانند مسیریابی ایمن و بهینه را تولید کنند، بسیار حائز اهمیت هستند [۱].

الگوریتم‌های برنامه‌ریزی مسیر را می‌توان به‌طور کلی به دودسته تقسیم کرد: الگوریتم‌های مبتنی بر جستجوی گراف [۳] و الگوریتم‌های مبتنی بر نمونه‌برداری [۷]. در الگوریتم‌های مبتنی بر جستجوی گراف، مانند الگوریتم اصلاح‌شده A^* ، فضای پیکربندی به یک گراف گسسته تبدیل شده و از الگوریتم‌های جستجوی گراف برای یافتن مسیر استفاده می‌شود. این الگوریتم‌ها کامل هستند، به این معنا که در صورت وجود مسیر، مسیر تعیین خواهد شد. با این حال، با افزایش ابعاد فضای پیکربندی، پیچیدگی محاسباتی آن‌ها به طور تصاعدی افزایش می‌یابد و برای برنامه‌ریزی مسیر بی‌درنگ در محیط‌های پیچیده نامناسب هستند [۸]. به‌طور کلی روش‌های سنتی برنامه‌ریزی مسیر، مانند الگوریتم‌های مبتنی بر شبکه A^* [۶، ۹]، Dijkstra و روش‌های مبتنی بر گراف، معمولاً در محیط‌های ایستا با مساحت کم عملکرد خوبی دارند [۱، ۲، ۱۰]. این الگوریتم‌ها در محیط‌های پویا و مساحت زیاد با محدودیت‌های اساسی روبه‌رو می‌شوند. الگوریتم A^* ، اگرچه در محیط‌های کوچک کارآمد است، اما با افزایش مقیاس محیط، به دلیل نیاز به پیمایش تعداد زیادی گره، قابلیت اجرای زمان واقعی ندارد، بنابراین برای کاربردهایی که نیاز به تصمیم‌گیری سریع و آنی دارند، مانند وسایل نقلیه خودمختار در محیط‌های شهری یا صنعتی، نامناسب است. علاوه

بر این، وابستگی این الگوریتم‌ها به نقشه‌های ایستا، باعث افزایش هزینه محاسباتی هنگام به‌روزرسانی مسیر در محیط‌های پویا می‌شود [۲، ۱۱]. برای رفع چالش‌های مذکور، الگوریتم‌های مسیریابی هندسی می‌توانند عملکرد مناسبی ارائه دهند [۱۲-۱۴]. این الگوریتم‌ها به دلیل سادگی محاسبات و قابلیت عملکرد سریع در محیط‌های پویا، در حال جایگزینی الگوریتم‌های پیچیده‌تر مانند الگوریتم‌های مبتنی بر جستجوی گراف و نمونه‌برداری هستند.

ازسوی دیگر، الگوریتم‌های مبتنی بر نمونه‌برداری، مانند RRT [۱۵]، به طور تصادفی فضای پیکربندی را نمونه‌برداری و یک نقشه راه احتمالی از مسیرهای بدون برخورد ایجاد می‌کنند. این الگوریتم‌ها از نظر محاسباتی کارآمدتر هستند و می‌توانند فضاهای پیکربندی با ابعاد بالا را مدیریت کنند. با این حال، آن‌ها از نظر احتمالی کامل هستند، به این معنی که ممکن است مسیر بهینه را پیدا نکنند یا اصلاً مسیری پیدا نکنند، به‌خصوص در محیط‌های پیچیده با موانع متعدد.

برای پرداختن به محدودیت‌های الگوریتم‌های سنتی برنامه‌ریزی مسیر، محققان الگوریتم‌های ترکیبی [۹، ۱۳] را بررسی کرده‌اند که در آن مزایای الگوریتم‌های مبتنی بر جستجوی گراف و مبتنی بر نمونه‌برداری را ترکیب می‌شود. به‌عنوان مثال، الگوریتم TargetTree-RRT [۱۶] با ادغام یک درخت هدف از پیش تعریف شده در الگوریتم RRT، برنامه‌ریزی مسیر پیوسته‌انحنا را برای پارکینگ خودران در محیط‌های پیچیده امکان‌پذیر می‌کند [۱۶]. به طور مشابه، الگوریتم LTA^* [۱۳] با ترکیب یک رویکرد مماس محلی با الگوریتم A^* ، برنامه‌ریزی مسیر بهینه را در نقشه‌های غیرمحدب تسهیل می‌کند که چالش این الگوریتم‌ها نیز عدم امکان اجرا به‌صورت بلادرنگ در مقیاس بالا است.

علاوه بر این، الگوریتم‌های برنامه‌ریزی مسیر برای کاربردهای خاص رباتیک در محیط‌های شهری سه‌بعدی یا ربات‌های متصل در فضاهای پیکربندی پیچیده، توسعه داده شده‌اند. به‌عنوان مثال، الگوریتم ۳D-TG با ساخت یک گراف مماس برای تولید سه‌زیر مسیر برای دورزدن موانع، برنامه‌ریزی مسیر کارآمد را برای پهپادها در محیط‌های شهری سه‌بعدی امکان‌پذیر می‌کند [۱۲] که همچنان از معایب روش مبتنی بر گراف بهره‌مند است.

همچنین، الگوریتم‌های برنامه‌ریزی مسیر برای سیستم‌های چند رباتی که در آن چندین ربات باید به طور مشترک در یک محیط حرکت کنند، توسعه داده شده‌اند. به‌عنوان مثال، الگوریتم SCOPP [۱۷] با تقسیم فضای کاری به زیر مناطق و تخصیص وظایف به ربات‌ها، برنامه‌ریزی مسیر پوشش مقیاس‌پذیر را برای سیستم‌های چند رباتی امکان‌پذیر می‌کند که لزوم امکان اجرای الگوریتم به‌صورت بلادرنگ در محیط‌های با مساحت بالا را مطرح می‌نماید. روش‌های مبتنی بر یادگیری تقویتی با چندین چالش روبه‌رو است که یکی از آن‌ها برآورده شدن قیود ایمنی است، به

^۱ Path planning

محیطی و تعامل میان رباط، موانع و هدف است که الگوریتم باید به طور برخط به آن‌ها پاسخ دهد.

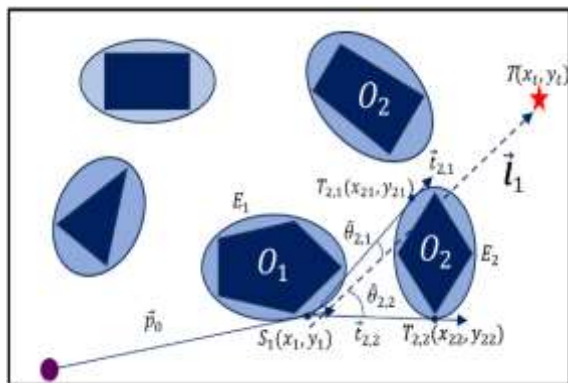
محیط مسیریابی به صورت دوبعدی مطابق شکل ۱ مدل سازی شده است که شامل مجموعه‌ای از موانع محدب، ربات و هدف است. ربات در موقعیت مشخص s قرار دارد و هدف آن این است که خود را به مختصات هدف متحرک که با T نشان داده می‌شود، برساند. مختصات موانع و موقعیت هدف دائماً در دسترس ربات قرار دارد.

در شکل ۱، هر مانع O_i به صورت مجموعه‌ای از نقاط تعریف شده است که محدوده آن را مشخص می‌کند. به جهت بهبود دقت و کارایی در شبیه سازی مسیر ربات، به جای استفاده از اشکال پیچیده موانع، از بیضی‌ها برای احاطه هر مانع استفاده می‌شود. این انتخاب به دلیل خصوصیات هندسی بیضی‌ها است که به صورت مطلوب فضای اطراف هر مانع را پوشش می‌دهد و در عین حال محاسبات بهینه سازی را ساده تر می‌کند. بیضی‌ها می‌توانند به طور مؤثری محدوده‌های غیرمنظم موانع را با دقت مناسب و به صورت بهینه محاط کنند. این روش باعث می‌شود تا الگوریتم مسیریابی قادر باشد به سرعت و با دقت به تغییرات محیطی واکنش نشان دهد.

در محاسبات مسیر ربات، خطوط مماس از موقعیت ربات به موانع ترسیم می‌شوند که مسیر امن و بدون برخورد محاسبه شود. این فرایند به الگوریتم کمک می‌کند تا به طور مؤثر از موانع دوری کرده و مسیر بهینه برای رسیدن به هدف مشخص کند.

الگوریتم مسیریابی باید توانایی انجام محاسبات در زمان واقعی را داشته باشد که به طور مؤثر به تغییرات محیطی پاسخ دهد. این تغییرات ممکن است شامل اضافه شدن موانع جدید به محیط یا تغییر موقعیت هدف باشد. برای مثال، اگر مانعی به محیط اضافه می‌شود، الگوریتم باید قادر باشد مسیر جدیدی پیدا کند که از برخورد با این مانع جلوگیری کند.

در ادامه روابط و نحوه تعیین بیضی محاطی بهینه و خطوط مماس بر آن که بیانگر مسیر حرکت ربات است ارائه خواهد شد.



شکل ۱: نمونه‌ای از محیط مسئله با پنج مانع محدب و موقعیت اولیه ربات و هدف

دلیل احتمال برخورد عوامل با یکدیگر و موانع تصمیم ایمنی یا همان ارضای قید عدم برخورد در آن ضروری است [۱۸]. روش میدان پتانسیل ممکن است در مینیم‌های محلی گیر کند و نیازمند تنظیم دقیق ابرپارامترها برای عملکرد مناسب است [۱۹]. روش کلاسیک Tangent Bug به دلیل انجام مسیریابی به صورت کور، توانایی یافتن مسیر بهینه و عبور امن از محیط‌های پیچیده را ندارد [۲۰].

جهت رفع این چالش‌های ذکر شده، الگوریتم هندسی جدید برای مسیریابی برخط در محیط‌های دوبعدی با موانع محدب معرفی شده است. این الگوریتم از مفاهیم خطوط مماس و بیضی‌های محاطی برای بهینه سازی مسیرها استفاده می‌کند و قادر است در محیط‌های با مقیاس بزرگ و پویا به طور مؤثر عمل کند [۲۱]. این روش، با کاهش پیچیدگی محاسباتی و تمرکز بر هندسه مسائل، امکان مسیریابی سریع و کارآمد را فراهم می‌کند. الگوریتم هندسی پیشنهادی^۱ در مقایسه با روش‌های سنتی مانند RRT، A* و ACO زمان محاسباتی کمتری دارد و مسیرهای کوتاه‌تر و نرم‌تری را تولید می‌کند. این ویژگی‌ها به ویژه در محیط‌های با تراکم موانع بالا و مساحت‌های وسیع مشهود است. به طور کلی برای پیش‌بینی مسیر، استفاده از الگوریتم‌های تکاملی ممکن است مناسب نباشد؛ زیرا بهینه نیستند، یعنی اثباتی برای بهینگی آن‌ها موجود نیست و عموماً جواب نزدیک به بهینه تولید می‌کنند [۲۲]. در مرجع [۲۳] در مورد روش‌های بهینه سازی مسیر بحث شده که می‌توان در نظر گرفت در الگوریتم‌هایی که مسیریابی بهینه انجام نمی‌دهند، با تعریف تابع هزینه مسیر بهینه را بین مسیرهای یافته شده تعیین نمود یا مسیر زیر بهینه را اصلاح کرد. همچنین، توانایی به روزرسانی مسیر در مواجهه با تغییرات محیطی، الگوریتم را برای کاربردهای برخط مانند ربات‌های متحرک مناسب می‌سازد.

در بخش دوم مقاله به تعریف مسئله پرداخته و شرایط مسئله بیان شده است. در بخش سوم نحوه مدل سازی موانع با استفاده از بیضی بهینه محاطی و روابط حاکم بر مسیرهای مماس ارائه می‌شود. در بخش چهارم، الگوریتم مسیریابی بهینه برخط به همراه شبه کد الگوریتم شرح داده شده است. بخش پنجم به نتایج شبیه سازی اختصاص یافته و در بخش ششم، نتایج کلی پژوهش مورد بحث و نتیجه گیری قرار می‌گیرد.

۲- تحلیل محیط و شرایط مسئله

در این بخش، محیط و شرایط مسئله‌ای که الگوریتم مسیریابی باید در آن عمل کند، به طور دقیق تحلیل و توصیف می‌شود. این تحلیل به عنوان مقدمه‌ای بر طراحی الگوریتم، شامل ویژگی‌های

^۱ Tangent Based Ellipse (ETB)

است. به منظور تعیین زاویه چرخش بیضی محاطی بهینه ابتدا مؤلفه‌های دوم مرکزی مطابق رابطه (۵) تعیین و زاویه چرخش مطابق رابطه (۶) تعیین خواهد شد. این بیضی‌ها نه تنها به عنوان یک تقریب هندسی برای مانع عمل کرده، بلکه به عنوان مبنای تعیین مسیر بهینه و امن برای ربات مورد استفاده قرار می‌گیرند. رابطه بیضی‌ها را می‌توان مطابق (۷) نوشت.

$$(7)$$

$$\mathbf{x}^T \mathbf{Q}_i \mathbf{x} + 2\mathbf{b}_i^T \mathbf{x} + c_i = 0$$

که در آن:

$$\begin{aligned} \bullet \quad \mathbf{R}_\varphi &= \begin{bmatrix} \cos(\varphi) & -\sin(\varphi) \\ \sin(\varphi) & \cos(\varphi) \end{bmatrix} \text{ ماتریس دوران،} \\ \bullet \quad \mathbf{Q}_i &= \mathbf{R}_\varphi \begin{bmatrix} \frac{1}{a_i^2} & 0 \\ 0 & \frac{1}{b_i^2} \end{bmatrix} \mathbf{R}_\varphi^T \text{ شامل مقادیر شعاع‌ها،} \\ \bullet \quad \mathbf{b}_i^T &= \left[\frac{-x_{ci}}{a_i^2}, \frac{-y_{ci}}{b_i^2} \right]^T \\ \bullet \quad \mathbf{x} &= [x, y] \\ \bullet \quad c_i &= \frac{x_{ci}^2}{a_i^2} + \frac{y_{ci}^2}{b_i^2} + 1 \text{ است.} \end{aligned}$$

معادله (۷) به طور کامل موقعیت موانع بیضی را، در فضای دوبعدی تعریف کرده و یکی از اجزای اساسی در تحلیل محیط و طراحی الگوریتم مسیریابی است.

۳-۱- روابط هندسی مسیر مماس بر موانع

در ادامه روابط هندسی جهت رسم مماس‌های $\vec{t}_{ij}$ که $j=1,2$ است ارائه خواهد شد (شکل ۲). به این منظور در نقاط مماسی $T_{ij}(x_{ij}, y_{ij})$ از رابطه بیضی (رابطه (۷)) گرادیان گرفته خواهد شد.

$$\nabla f(\mathbf{x}) = 2\mathbf{Q}_i \mathbf{x} + \mathbf{b}_i \quad (8)$$

معادله خط در صفحه دوبعدی مطابق (۹) خواهد بود.

$$(9)$$

که در آن $\mathbf{n} = [n_x, n_y]^T$ شیب خط، d مقدار ثابت و $\mathbf{x} = [x, y]^T$ است. در نقطه مماسی شیب خط مماس با مقدار گرادیان برابر است و می‌توان رابطه (۱۰) را به شکل مقابل نوشت که در آن

$$\mathbf{p}_{T_{ij}} = (x_{ij}, y_{ij})^T$$

$$\mathbf{n}_{ij} = \nabla f(\mathbf{p}_{T_{ij}}) \quad (10)$$

جهت محاسبه نقطه مماسی از حل معادله رابطه (۱۱) و (۱۲) بهره گرفته خواهد شد. بر اساس این معادلات نقطه

$$T_{ij}(x_{ij}, y_{ij}) \text{ در رابطه خط مماس و نقطه مماسی } S_{i-1}(x_{i-1}, y_{i-1})$$

در معادله بیضی صدق می‌کند، بنابراین از حل این دو معادله نقاط مماسی تعیین خواهند شد.

$$\nabla f(\mathbf{p}_{T_{ij}})^T \mathbf{p}_{S_{i-1}} + d_{ij} = 0 \quad (11)$$

$$(12)$$

$$\mathbf{p}_{T_{ij}}^T \mathbf{Q}_i \mathbf{p}_{T_{ij}} + 2\mathbf{b}_i^T \mathbf{p}_{T_{ij}} + c_i = 0$$

که در آن $\mathbf{p}_{S_i} = (x_i, y_i)$ است.

۳- مدل سازی موانع با بیضی بهینه مماسی

در این بخش، به طور دقیق‌تر به بررسی نحوه یافتن بیضی محاطی بهینه برای موانع در محیط مسیریابی پرداخته می‌شود. این فرایند یک مسئله بهینه‌سازی است که هدف آن محاط کردن موانع در یک بیضی با کمترین انحراف از رأس آن‌ها است.

مختصات هر مانع با قید محدب بودن به صورت مجموعه‌ای از نقاط $\{(x_{i1}, y_{i1}), \dots, (x_{iK}, y_{iK})\}$ دریافت خواهد شد که i بیانگر شماره موانع و K تعداد رأس‌های مانع خواهد بود. جهت حل مسئله بهینه‌سازی و تعیین بیضی محاطی بهینه $C_{Ei}(x_{ci}, y_{ci})$ از (۱) جهت تعیین مرکز بیضی و (۲) جهت تعیین شعاع کوچک و بزرگ بیضی بهره گرفته می‌شود. به منظور افزایش ایمنی در هنگام عبور از مجاورت موانع، پارامتر δ به شعاع بیضی اضافه می‌شود تا فاصله‌ی ایمن در برابر برخورد تضمین گردد. این سازوکار علاوه بر حفظ ایمنی، موجب ایجاد رفتاری مشابه حرکت ساحلی^۱ نیز می‌شود که در فرایند موقعیت‌یابی و مسیریابی، یک مزیت محسوب می‌شود. مقداری δ با توجه به ویژگی‌های دینامیکی ربات (نظیر شعاع گردش و سرعت واکنش) قابل تنظیم است؛ به عنوان مثال، ربات‌هایی که توانایی چرخش کمتری دارند نیازمند مقادیر بزرگ‌تر δ هستند. به عنوان مثال در رابطه (۳) میزان δ برابر با ۱۰ درصد شعاع مانع در نظر گرفته شده است.

$$(1)$$

$$x_{ci} = \frac{1}{n} \sum_{k=1}^K x_k, \quad y_{ci} = \frac{1}{n} \sum_{k=1}^K y_k \quad (2)$$

$$\mathbf{P}_i = \begin{bmatrix} \text{var}(x_{ci}) & \text{cov}(x_{ci}, y_{ci}) \\ \text{cov}(x_{ci}, y_{ci}) & \text{var}(y_{ci}) \end{bmatrix}$$

$$\delta_1 = 0.1 \times \sqrt{\lambda_1}, \quad \delta_2 = 0.1 \times \sqrt{\lambda_2} \quad (3)$$

$$a_i = \sqrt{\lambda_1} + \delta_1, \quad b_i = \sqrt{\lambda_2} + \delta_2$$

$$(4)$$

$$(5)$$

$$\begin{aligned} S_{xx} &= \frac{1}{n} \sum_{k=1}^K (x_k - x_{ci})^2, \quad S_{yy} = \frac{1}{n} \sum_{k=1}^K (y_k - y_{ci})^2 \\ S_{xy} &= \frac{1}{n} \sum_{k=1}^K (x_k - x_{ci})(y_k - y_{ci}) \end{aligned} \quad (6)$$

$$\phi = \frac{1}{\gamma} \text{atan2}(\gamma S_{xy}, S_{xx} - S_{yy})$$

که λ_1 و λ_2 بزرگ‌ترین مقادیر ویژه اول و دوم ماتریس $\mathbf{P}_i$ هستند. با استفاده از رابطه (۲) برای هر مانع محدب، بیضی‌ای محاط می‌شود که به طور دقیق نمایانگر مرزهای آن است. مطابق شکل ۱ هر مانع با بیضی گون E_i با مختصات مبدأ $C_{Ei}(x_{ci}, y_{ci})$ شعاع بزرگ a_i و شعاع کوچک b_i (روابط (۲) تا (۴)) محاط شده

^۱ Coastal Behavior

^۲ Maneuvring Capacity

$$(20)$$

$$J_0 = \sum_{p_i}^{p_i} \theta_i$$

$$J_0^{\text{norm}} = \frac{J_0}{\pi} \quad (21)$$

مؤلفه سوم مطابق (۲۲) تعیین می‌شود که در آن θ_i در هر تکه مسیر انتخاب شده θ_i در صورت وجود مانع برابر با یک و در غیر این صورت برابر با صفر خواهد بود. به عبارتی این مؤلفه بیانگر مجموع تعداد موانع در مسیر نهایی است. این مؤلفه مطابق رابطه (۲۳) و با تقسیم بر تعداد کل موانع موجود در محیط θ_T استاندارد می‌شود.

$$J_0 = \sum_{p_i}^{p_i} \theta_i \quad (22)$$

$$J_0^{\text{norm}} = \frac{J_0}{\theta_T} \quad (23)$$

در رابطه (۱۷) ضرایب ω_0 ، ω_d و ω_θ وزن‌های تابع هزینه هستند که مجموع آن‌ها برابر یک است. مقدار این ضرایب میزان اهمیت مؤلفه متناظر در تابع هزینه را تعیین می‌کند و مطابق با نیاز طراح انتخاب می‌شود. به عنوان مثال، در کاربردهایی که ایمنی اهمیت بالاتری دارد، وزن مربوط به مانع افزایش می‌یابد؛ درحالی‌که در کاربردهایی که کوتاه‌بودن مسیر مهم‌تر است، وزن طول مسیر افزایش خواهد یافت.

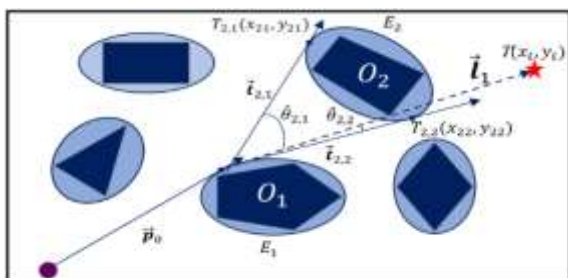
در نهایت از طریق حل مسئله بهینه‌سازی رابطه (۲۴) که در آن مقدار m از رابطه (۱۶) تعیین می‌شود، مسیری که تابع هزینه را کمینه کند تعیین خواهد شد.

$$(24)$$

$$\min_{p_1, p_2, \dots, p_m} J_{\text{total}}$$

تابع هزینه رابطه (۲۴) دارای مؤلفه‌های محدب (روابط (۱۹)، (۲۱) و (۲۳)) است. از آنجاکه حرکت ربات در هر گام تنها در یک ناحیه محدود و بدون مانع (دامنه محدب) بررسی می‌شود و مسیر به صورت سلسله‌مراتبی و گام‌به‌گام انتخاب می‌گردد، مسئله در هر مرحله یک مسئله بهینه‌سازی محدب خواهد بود. برای مسائل محدب، حل بهینه محلی همان بهینه جهانی است؛ بنابراین مسیر حاصل در هر گام از دید ریاضی بهینه خواهد بود [۲۴].

مراحل اجرای الگوریتم به صورت دیاگرام و شبه‌کد در ادامه ذکر شده است. در شکل ۲ و شکل ۴ مراحل رسم مماس به ازای دو مانع در مسیر قابل مشاهده است. همچنین در شکل ۳ و شکل ۵ نیز به ترتیب دو مسیر ممکن از چهار مسیر ممکن نمایش داده شده است.



شکل ۲: رسم خطوط مماسی از نقطه مماسی اول مانع اول به مانع دوم

بنابراین، از حل معادلات (۱۱) و (۱۲) و بهره‌گیری از رابطه (۱۰) شیب خطوط مماس بر هر مانع محاسبه خواهد شد. جهت تعیین زوایای θ_{ij} که زاویه بین خطوط مماس و خط مستقیم $\vec{L}_{i-1}$ هستند می‌توان از رابطه (۱۳) بهره گرفت.

$$(13)$$

$$\theta_{ij} = \tan^{-1} \left(\frac{m_{ij} - m_{i-1}}{1 + m_{ij} m_{i-1}} \right) \quad (14)$$

$$\theta_{ij} \in \left(-\frac{\pi}{2}, \frac{\pi}{2} \right)$$

نظر به ضرورت بهینه بودن مسیر، زاویه سمت^۱ حرکت ربات همواره در راستای هدف در نظر گرفته می‌شود؛ بنابراین زاویه رابطه (۱۳) بین $(-\frac{\pi}{2}, \frac{\pi}{2})$ در نظر گرفته شده است. در نهایت جهت اطمینان از اینکه مماس حاصل با سایر موانع تقاطع دارد یا خیر، باید معادله مماس در معادله موانع جای‌گذاری شود (معادله (۱۵)). اگر معادله حاصل یک ریشه داشته باشد، آن‌گاه مماس به مانع جدید نیز مماس خواهد بود. اما اگر دو ریشه داشته باشد، خط مماس از مانع عبور می‌کند و باید مانع در نظر گرفته شود، در غیر این صورت در مسیر مماس مانعی نخواهد بود.

$$(z+t.d)^T Q (z+t.d) + b^T (z+t.d) + c = 0 \quad (15)$$

که در آن $(z+t.d)$ به عنوان پارامتری از خط است که موقعیت نقطه‌ای روی خط را با استفاده از پارامتر t تعیین می‌کند. این نقطه روی خط به طور پیوسته از نقطه شروع z در جهت بردار d حرکت می‌کند.

با توجه به تعداد موانع i در مسیر ربات، تعداد مسیرهای ممکن از رابطه (۱۶) قابل محاسبه است.

$$m = n(\text{Paths}) = 2^i \quad (16)$$

۴- طراحی برخط مسیر بهینه بدون برخورد با موانع

به منظور تعیین مسیر بهینه باید تابع هزینه رابطه (۱۷) کمینه شود، در این رابطه سه مؤلفه استاندارد در نظر گرفته شده است. مؤلفه اول مطابق رابطه (۱۸) تعیین می‌شود که بیانگر مجموع طول مسیر طی شده ربات است که با استفاده از رابطه (۱۹) که در آن $\|\vec{L}_i\|$ بیانگر طول خط مستقیم از نقطه اولیه به هدف است، استانداردسازی خواهد شد.

$$J_{\text{total}} = \omega_d J_d^{\text{norm}} + \omega_\theta J_\theta^{\text{norm}} + \omega_0 J_0^{\text{norm}} \quad (17)$$

$$J_d = \sum_{k=1}^i \|\vec{p}_k\| \quad (18)$$

$$J_d^{\text{norm}} = \frac{J_d}{\|\vec{L}_i\|} \quad (19)$$

مؤلفه دوم مطابق (۲۰) تعیین می‌شود که در آن زوایای از رابطه (۱۳) محاسبه می‌شوند. این مؤلفه بیانگر مجموع انحراف از مسیر اصلی است که با استفاده از رابطه (۲۱) استاندارد می‌شود.

^۱ Heading Angel

شده و هزینه آن طبق گام ۵ محاسبه شود)

۵: - محاسبه مسیر بهینه:

در میان مسیرهای پیشنهادی: P برای هر مسیر ممکن محاسبه طول مسیر و استاندارد کردن مطابق روابط (۱۹-۱۸) محاسبه مجموع زاویه‌ها و استاندارد کردن زوایا مطابق روابط (۲۰-۲۱)

محاسبه تعداد موانع موجود در مسیر P و استاندارد کردن مطابق روابط (۲۲-۲۳)

حل مسئله بهینه‌سازی رابطه (۲۴) و تعیین مسیر بهینه

۶: به‌روزرسانی مسیر ربات به سمت هدف در مسیر انتخاب شده در صورت افزوده شدن موانع جدید و یا تغییر موقعیت هدف مراحل ۱-۵ تکرار شده و مسیر ربات به‌روز می‌شود.

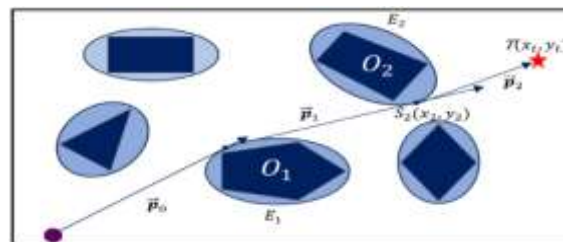
۵- نتایج شبیه‌سازی

شبیه‌سازی الگوریتم مسیریابی با استفاده از محاط شدن بیضی‌ها بر روی موانع یا بیضی به‌عنوان موانع با پیاده‌سازی الگوریتم پیشنهادی (ETB) در نرم‌افزار MATLAB انجام گرفت. برای ارزیابی عملکرد الگوریتم‌ها، دو محیط شبیه‌سازی مختلف طراحی شد. جزئیات مختصات نقاط شروع و پایان این محیط‌ها در جدول ۱ آورده شده است.

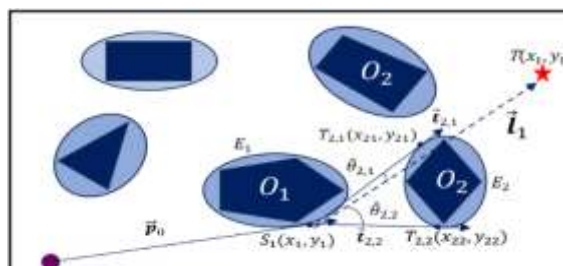
در این شبیه‌سازی، از واحدهای مقیاس دلخواه برای مدل‌سازی محیط استفاده شده است. این مقیاس به‌گونه‌ای انتخاب شده که درک بهتری از رفتار الگوریتم‌ها در شرایط مختلف محیطی را فراهم کرده و به‌راحتی می‌تواند به واحدهای واقعی (مانند متر یا سانتی‌متر) مقیاس‌بندی شود. به‌طور مثال، در این شبیه‌سازی هر واحد به‌طور تقریبی معادل ۱ متر در محیط واقعی در نظر گرفته شده است.

جدول ۱: مختصات و جزئیات محیط‌های شبیه‌سازی

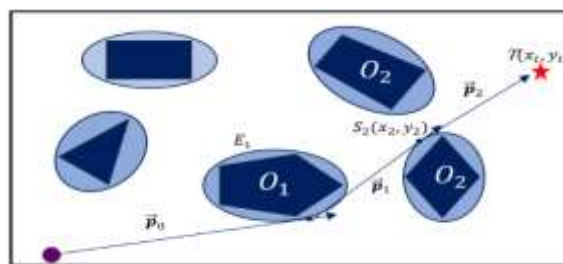
شبیه‌سازی محیط شماره ۱				شبیه‌سازی محیط شماره ۲			
مقیاس محیط: ۱۶۰۰ × ۱۰۰۰				مقیاس محیط: ۱۶۰۰ × ۱۰۰۰			
مختصات عامل		مختصات هدف		مختصات عامل		مختصات هدف	
(y,x)		(y,x)		(y,x)		(y,x)	
۰ ، ۰		۱۶۰۰ ، ۱۰۰۰		۰ ، ۰		۱۶۰۰ ، ۱۰۰۰	
شماره	مختصات مرکز (y,x)	شماره	مختصات مرکز (y,x)	شماره	مختصات مرکز (y,x)	شماره	مختصات مرکز (y,x)
ممانع	(r)	ممانع	(r)	ممانع	(r)	ممانع	(r)
۱	۵۶۳ ، ۸۴۲	۱	۷۴	۱	۴۶۱ ، ۴۲۷	۸۰	
۲	۱۱۵ ، ۵۱۲	۲	۶۷	۲	۱۴۰ ، ۸۴۴	۸۷	
۳	۹۲۵ ، ۶۵۴	۳	۶۴	۳	۷۸۴ ، ۵۶۳	۶۲	
۴	۱۳۹۴ ، ۴۱۳	۴	۸۹	۴	۶۱۴ ، ۸۹۱	۵۱	
۵	۸۵۹ ، ۲۹۳	۵	۹۵	۵	۱۳۴۰ ، ۸۳۱	۹۰	



شکل ۳: مسیر تعیین شده به‌ازای انتخاب مماس اول مانع اول و مماس دوم مانع دوم



شکل ۴: رسم خطوط مماسی از نقطه مماسی دوم مانع اول به مانع دوم



شکل ۵: مسیر تعیین شده به‌ازای انتخاب مماس دوم مانع اول و مماس اول مانع دوم

الگوریتم برنامه‌ریزی مسیر ETB

- ۱: $S \leftarrow$ مختصات موقعیت شروع ربات
 $T \leftarrow$ مختصات موقعیت هدف
 برای هر مانع O_i که مجموعه‌ای از نقاط $\{(x_{i1}, y_{i1}), \dots, (x_{ik}, y_{ik})\}$ است:
 محاسبه مرکز بیضی $C_{Ei}(x_{Ci}, y_{Ci})$ برای هر مانع مطابق رابطه (۱)
 محاسبه شعاع‌های بزرگ و کوچک بیضی برای هر مانع مطابق رابطه (۲)
 در هر گام مسیریابی:
 - ۳: بررسی شود که آیا ربات به نقطه هدف نزدیک شده است یا خیر:
 اگر فاصله ربات از نقطه هدف بزرگ‌تر یا برابر با ϵ است مسیر را تکمیل کنید و الگوریتم خاتمه یابد. مقدار ϵ با توجه به میزان خطای ربات در قرارگیری و حفظ موقعیت در نقطه هدف تعیین شود.
 - در غیر این صورت، مسیر بهینه محاسبه شود:
 - محاسبه مماس‌ها: برای هر مانع O_i :
 برای هر مانع O_i ، دو مماس مطابق روابط (۱۲-۸) محاسبه شود.
 برای هر مماس T_{ij} : محاسبه زاویه θ_{ij} بین خط مماس و مسیر فعلی ربات مطابق رابطه (۱۳)
 بررسی کنید که آیا خط مماس با موانع دیگر تقاطع دارد یا خیر
 اگر تقاطع دارد مسیر اصلاح شود گام ۳ را با مانعی که با مسیر تقاطع داشته است انجام دهید. (مسیر مماس دیگر نیز بررسی

سرعت ثابت	۱۰	m/s
-----------	----	-----

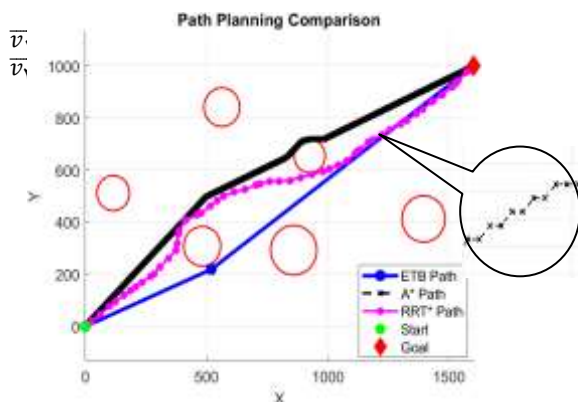
۶	۴۸۲ ، ۳۰۸	۷۶	۶	۱۰۵۲ ، ۲۰۹	۸۶
---	-----------	----	---	------------	----

مقایسه مسیرهای الگوریتم‌ها

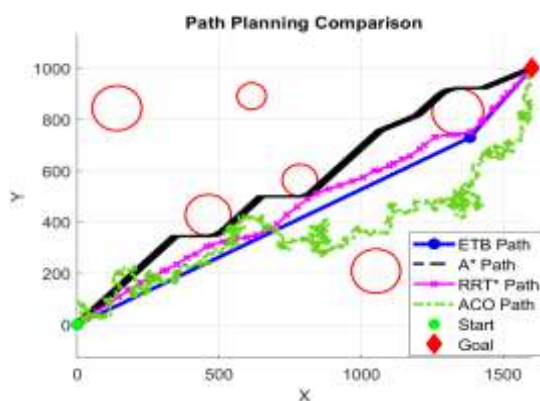
در شکل ۶ و شکل ۷، مسیرهای تولید شده توسط الگوریتم ETB در مقایسه با الگوریتم‌های A*، RRT* و ACO نشان داده شده است. در این نمودارها، مشخص می‌شود که الگوریتم ETB به دلیل استفاده از هندسه مماسی و مدل‌سازی بیضوی موانع، توانسته است مسیرهایی کوتاه‌تر، صاف‌تر و با هزینه زمانی و محاسباتی کمتر ایجاد کند.

همان‌طور که مشاهده می‌شود، الگوریتم ETB مسیرهایی را تولید کرده است که نه تنها کوتاه‌تر از سایر الگوریتم‌ها هستند، بلکه از نرمی و انحنای کمتری برخوردارند. همچنین، تمرکز بر مقایسه دقیق‌تر مسیرها است.

نتایج مسیرهای محاسبه شده توسط معیارهای ارزیابی عملکرد الگوریتم پیشنهادی با روش‌های مرسوم نظیر A*، RRT، RRT* و ACO در جدول ۳ مقایسه گردید.



شکل ۶: مقایسه مسیرهای تولید شده توسط الگوریتم‌های مختلف در محیط شماره ۱



شکل ۷: مقایسه مسیرهای تولید شده توسط الگوریتم‌های مختلف

تحلیل نتایج

معیارهای ارزیابی عملکرد

برای تحلیل عملکرد الگوریتم‌ها، از معیارهای مختلفی برای ارزیابی استفاده شد که شامل طول مسیر، زمان اجرا، نرمی مسیر، مصرف انرژی و تعداد نقاط مسیر تولید شده است. این معیارها به طور دقیق به شرح زیر تعریف شده‌اند:

طول مسیر: مجموع مسافت طی شده توسط ربات از نقطه شروع تا مقصد.

زمان اجرا: مدت زمانی که ربات برای مسیریابی از نقطه شروع تا مقصد با استفاده از الگوریتم مشخص به‌طور کلی نیاز دارد.

نرمی مسیر: میزان تغییرات زاویه‌ای در مسیر طی شده که به‌عنوان معیاری برای پیچیدگی مسیر استفاده می‌شود.

برای محاسبه تغییرات زاویه‌ای بین نقاط متوالی مسیر محاسبه می‌شود. برای هر سه نقطه متوالی (p_{i-1}, p_i, p_{i+1}) ، زاویه تغییر مسیر به‌صورت رابطه (۲۵) محاسبه می‌شود:

$$\theta_i = \cos^{-1} \left(\frac{\vec{p}_1 \cdot \vec{p}_2}{\|\vec{p}_1\| \cdot \|\vec{p}_2\|} \right) \quad (25)$$

که در آن:

نرمی مسیر به‌عنوان مجموع قدرمطلق این زاویه‌ها در تمام مسیر به‌صورت رابطه (۲۶) محاسبه می‌شود.

$$\text{Smoothness} = \sum_{i=1}^{n-1} |\theta_i| \quad (26)$$

تعداد نقاط مسیر تولید شده: عداد نقاط موردنیاز برای رسیدن به هدف.

مصرف انرژی: برای محاسبه مصرف انرژی ربات در طول مسیر از مدل انرژی مصرفی ربات که مشخصات آن در جدول ۲ ذکر شده است، استفاده گردید.

جدول ۲: مشخصات ربات برای محاسبه مصرف انرژی

پارامتر	مقدار	واحد
جرم ربات	۲	kg
ضریب مقاومت غلتشی	۰/۰۵	-
ضریب درگ	۰/۰۳	m ²
چگالی هوا	۱/۲	kg/m ³
بازده محرکه	۰/۶۵	-
بازده بازیابی انرژی	۰	-
شتاب گرانشی	۹/۸۱	m/s ²

نتایج جدول ۳ نشان می‌دهند که الگوریتم ETB به طور قابل توجهی از نظر زمان محاسباتی و نرمی مسیر بر سایر الگوریتم‌ها برتری دارد. به عنوان مثال، در محیط شبیه‌سازی شماره ۱، زمان اجرای ETB تنها ۰/۰۳۲ ثانیه است، در حالی که A^* و RRT^* به ترتیب ۳/۹۸۱ ثانیه و ۴/۱ ثانیه زمان نیاز دارند. این کاهش چشمگیر در زمان محاسبات، نشان‌دهنده بهره‌وری بالای الگوریتم ETB در پردازش برخط مسیرها است. الگوریتم پیشنهادی به دلیل بهره‌گیری از اصول هندسی و عدم نیاز به محاسبات گرافی گسترده، زمان محاسباتی بسیار کمتری نسبت به روش‌های مرسوم دارد به‌ویژه در محیط‌های با مساحت زیاد که بار محاسباتی الگوریتم‌های مرسوم بسیار بیشتر از روش هندسی است. این ویژگی به‌ویژه برای کاربردهای برخط، نظیر ربات‌های متحرک، بسیار حائز اهمیت است. در ۱۵۰ شبیه‌سازی با محیط‌های مختلف برای انجام مقایسه با الگوریتم A^* بادقت شبکه‌کردن ۱ نقشه و RRT^* با اندازه گام ۵ متر با شعاع همسایگی ۵۰ متر، الگوریتم هندسی مبتنی بر مماس در به‌صورت میانگین در ۰/۰۲۶ ثانیه و الگوریتم A^* در ۴/۲۳ ثانیه مسیریابی را محاسبه نمود.

جدول ۳: مقایسه کیمیت‌های مختلف ارزیابی عملکرد برای الگوریتم‌های مختلف در دو محیط شبیه‌سازی

ACO	RRT*	A*	ETB	کمیت مقایسه	گروه
-	۴/۱	۳/۹۸۱	۰/۰۳۲	زمان اجرا (s)	۱
-	۱۹۵۲/۲	۱۱۴۵۱/۵	۱۸۹۶/۹	طول مسیر (m)	
-	۵/۱۴۷	۶۹۶/۶	۰/۲۲۵	نرمی مسیر	
-	۸۳۵۲/۳۹	۸۶۱۱/۶۸	۸۱۱۵/۶۲	مصرف انرژی (j)	
-	۵۹	۱۶۰۰	۳	تعداد نقاط مسیر	
۲۸۹۹/۲۸	۴/۹۲۷	۳/۲۷۷	۰/۱۲۴	زمان اجرا (s)	۲
۱۴۶۴۱/۵	۱۹۲۹/۱	۱۲۱۰۳/۲	۱۹۱۰/۳	طول مسیر (m)	
۲۱۰۸/۹	۵/۸۳۴	۲۸۹/۸۱	۰/۴۰	نرمی مسیر	
۶۲۶۴۳/۲۸	۸۲۵۳/۶۱	۸۶۱۱/۶۸	۸۱۷۳/۲۴	مصرف انرژی (j)	
۱۴۶۶	۵۳	۱۶۰۰	۳	تعداد نقاط مسیر	

شکل ۸ به‌روزرسانی مسیر در زمان‌های مختلف را نشان می‌دهد. در این شبیه‌سازی، ربات با تغییر موقعیت موانع و هدف، مسیر جدید خود را در کمترین زمان ممکن محاسبه کرده و به سمت هدف جدید حرکت می‌کند. این ویژگی الگوریتم ETB را به گزینه‌ای ایدئال برای محیط‌های پویا تبدیل می‌کند.

ارزیابی ضرایب تابع هزینه

به‌منظور ارزیابی تأثیر ضرایب تابع هزینه بر عملکرد الگوریتم، سه حالت مختلف برای ضرایب تابع هزینه در نظر گرفته شد. این ضرایب شامل وزن‌های طول مسیر (ω_d)، زاویه مسیر (ω_θ) و موانع (ω_o) هستند که در رابطه (۱۷) به‌طور کلی به‌صورت زیر تعریف شده‌اند:

در حالت اول، ضرایب تابع هزینه (J_1) به‌صورت زیر تنظیم شدند: ($\omega_d = 0.3, \omega_\theta = 0, \omega_o = 0.7$) که وزن طول مسیر (ω_d) طول مسیر بیشتر از سایر مؤلفه‌ها تنظیم شد تا الگوریتم بیشتر به کوتاه‌تر کردن مسیر و کمترین تعداد مانع متمایل شود. در این حالت، انتظار می‌رود که الگوریتم مسیری کوتاه‌تر ایجاد کند.

در حالت دوم، ضرایب تابع هزینه (J_2) به‌صورت زیر تنظیم شدند: ($\omega_d = 0.3, \omega_\theta = 0.6, \omega_o = 0.1$) وزن زاویه مسیر (ω_θ) بیشتر از سایر مؤلفه‌ها در نظر گرفته شد تا الگوریتم مسیرهایی با زاویه کمتر نسبت به هدف پیدا کند...

در حالت سوم، ضرایب تابع هزینه (J_3) به‌صورت زیر تنظیم شدند: ($\omega_d = 0.45, \omega_\theta = 0.45, \omega_o = 0.1$)

در شکل ۹، نتایج به‌دست‌آمده از سه حالت مختلف ضرایب تابع هزینه به‌طور دقیق مقایسه شده‌اند و نشان می‌دهند که تغییر ضرایب تابع هزینه چگونه می‌تواند تأثیرات متفاوتی بر طول مسیر، نرمی و سایر پارامترهای مربوطه داشته باشد.

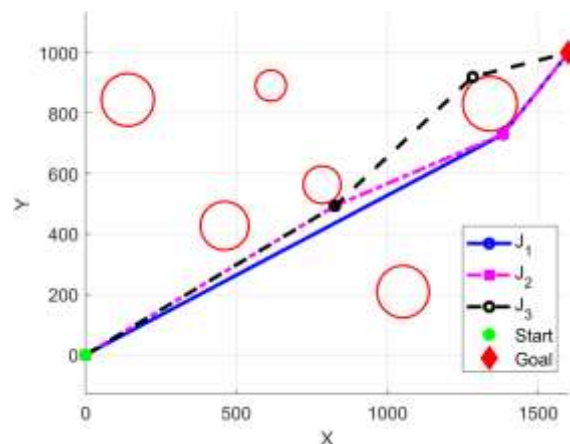
از منظر طول مسیر، ETB مسیرهای کوتاه‌تری نسبت به RRT* و ACO ارائه می‌دهد و با کاهش تعداد نقاط مسیر، پیچیدگی مسیر کاهش‌یافته و نرمی مسیر به شکل قابل‌توجهی افزایش‌یافته است. به‌عنوان نمونه، در محیط شبیه‌سازی ۱، نرمی مسیر ETB برابر ۰/۲۲۵ است درحالی‌که این مقدار برای A* به ۶۹۶/۶ و برای RRT* به ۵/۱۵۷ می‌رسد. این نشان می‌دهد که مسیرهای تولید شده توسط ETB صاف‌تر و با تغییرات زاویه‌ای کمتر هستند که به کاهش مصرف انرژی و کارایی بالاتر ربات کمک می‌کند.

در محیط شبیه‌سازی شماره ۲، الگوریتم ETB همچنان عملکرد بهینه‌ای ارائه می‌دهد. اگرچه زمان اجرا کمی افزایش‌یافته است (۰/۱۲۴ ثانیه)، اما نرمی مسیر (۰/۴) و تعداد نقاط مسیر نشان می‌دهند که الگوریتم توانسته است مسیرهای کوتاه و نرم با حداقل نقاط مسیر تولید کند. در همین محیط، الگوریتم‌های A*، RRT* و ACO مسیرهایی با طول بیشتر، نرمی کمتر یا تعداد نقاط مسیر بسیار بیشتر ایجاد کردند که نشان‌دهنده افزایش پیچیدگی محاسباتی و انرژی مصرفی بالاتر است. مشاهده معیار مصرف انرژی نیز نشان‌دهنده کارایی بالای الگوریتم ETB است. این کاهش مصرف انرژی به دلیل انتخاب مسیرهای کوتاه و نرم با حداقل نقاط مسیر توسط الگوریتم ETB است.

مسیریابی در محیط‌های پویا

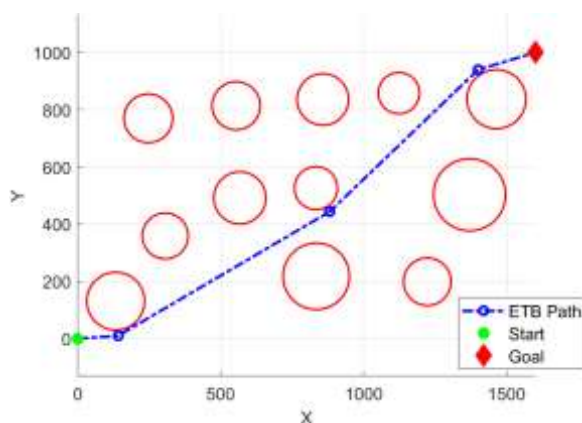
یکی از ویژگی‌های برجسته الگوریتم ETB، توانایی به‌روزرسانی مسیر در زمان واقعی است. در شکل ۸، تغییرات موقعیت هدف و موانع در طول زمان نشان‌داده شده است. این ویژگی به‌ویژه برای ربات‌ها در محیط‌های دینامیک اهمیت دارد، زیرا محیط در حال تغییر است و ربات باید بتواند مسیر خود را بر اساس تغییرات آن به‌طور برخط محاسبه کند.

افزایش پیچیدگی محیط و نیاز به انجام محاسبات بیشتر برای پیدا کردن مسیر بهینه است.



شکل ۹: برنامه‌ریزی مسیر محیط شبیه‌سازی شماره ۲ با توابع هزینه مختلف

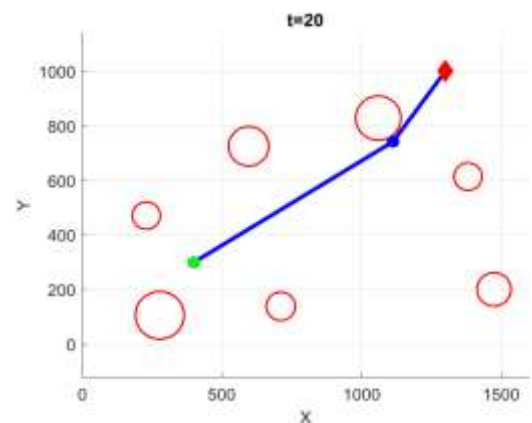
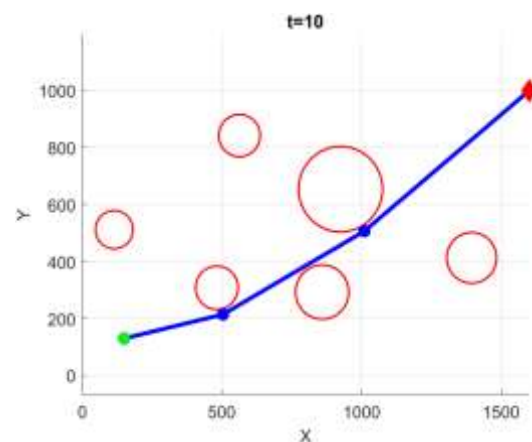
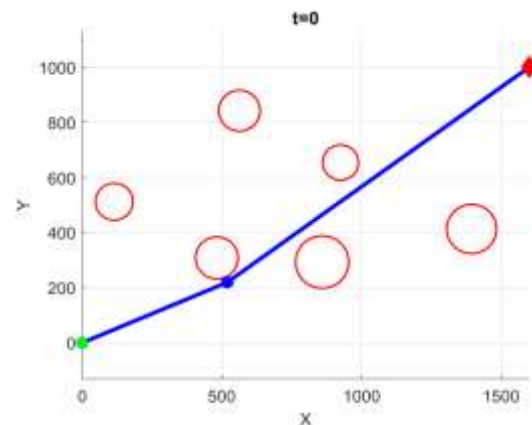
- طول مسیر با تابع هزینه (J_1) : ۱۹۱۰/۳
- طول مسیر با تابع هزینه (J_2) : ۱۹۱۳/۷
- طول مسیر با تابع هزینه (J_3) : ۱۹۱۴/۴



شکل ۱۰: عملکرد الگوریتم در محیط با تراکم زیاد موانع محیطی

جدول ۴: مقدار معیارهای عملکردی محاسبه شده در محیط با تراکم و تعداد بیشتر موانع محیطی

مقدار محاسبه شده	کمیت‌های محاسبه شده مسیر
۰/۰۹	زمان اجرا (s)
۱۹۲۵/۹	طول مسیر (m)
۱/۱۶۴	نرمی مسیر
۸۲۳۹/۹	مصرف انرژی (j)
۴	تعداد نقاط مسیر



شکل ۸: به‌روزرسانی مسیر در زمان‌های مختلف برای محیط اولیه شبیه‌سازی شماره ۱ و حرکت ربات

عملکرد الگوریتم در شرایط تراکم موانع

در شکل ۱۰، عملکرد الگوریتم در محیط‌های با تراکم بالاتر موانع بررسی شده است. نتایج شبیه‌سازی‌ها نشان می‌دهند که الگوریتم ETB توانسته است مسیرهای ایمن و بهینه پیدا کند و به‌خوبی با موانع زیاد و محیط‌های پیچیده سازگار شود. با افزایش تراکم موانع به تعداد ۱۲ عدد، زمان محاسباتی الگوریتم کمی افزایش یافته و نرمی مسیر کاهش می‌یابد. این امر به دلیل

۶- نتیجه گیری

در این پژوهش الگوریتم هندسی جدیدی با عنوان ETB برای مسیریابی برخط ربات‌های خودمختار در محیط‌های دوبعدی با موانع محدب معرفی شد. نوآوری اصلی این روش در ترکیب مدل‌سازی بیضوی موانع و استفاده از خطوط مماسی برای تولید مسیرهای کاندید است؛ رویکردی که مسئله را در هر گام به یک ناحیه‌ی محدب کاهش داده و بدین ترتیب بهینگی مرحله‌ای، کاهش هزینه محاسباتی و قابلیت به‌روزرسانی سریع مسیر در محیط‌های پویا را تضمین می‌کند.

نتایج شبیه‌سازی‌ها نشان داد که ETB نسبت به الگوریتم‌های مرسوم عملکردی برتر دارد. زمان محاسباتی به طور متوسط ۶۰ درصد کمتر از A^* و ۲۰ درصد کمتر از RRT^* بود، درحالی‌که مسیرها کوتاه‌تر و نرم‌تر تولید شدند (برای مثال نرمی مسیر ۰/۲۲۵ در برابر ۶۹۶/۶ در A^* و ۵/۱۴۷ در RRT^*) همچنین تعداد نقاط موردنیاز برای تعریف مسیر تنها سه‌نقطه بود که نسبت به بیش از ۱۶۰۰ نقطه در A^* کاهش چشمگیری محسوب می‌شود. علاوه بر این، مصرف انرژی نیز به طور میانگین ۵ تا ۱۰ درصد کمتر از الگوریتم‌های مقایسه شده محاسبه شده است. الگوریتم پیشنهادی در محیط‌های با تعداد زیاد موانع ممکن است با پیچیدگی بیشتری محاسباتی مواجه شود. بااین‌وجود، نتایج شبیه‌سازی‌ها نشان می‌دهد که این الگوریتم در یافتن مسیرهای بهینه و پاسخگویی سریع به تغییرات محیطی کارآمد است و پتانسیل بالایی برای بهبود و استفاده در تحقیقات آینده در زمینه مسیریابی ربات‌ها در محیط‌های سه‌بعدی و پیچیده دارد و با مدل کردن موانع غیرمحدب به‌صورت بیضی‌گون‌های محدب می‌توان در محیط با موانع غیرمحدب نیز بسط داده شود.

در مجموع، می‌توان نتیجه گرفت که ETB روشی ساده، سریع و مقیاس‌پذیر برای مسیریابی برخط در محیط‌های محدب پیچیده و پویاست که توانایی بالایی در بهبود کارایی ربات‌های متحرک و وسایل نقلیه خودران دارد. توسعه‌ی این الگوریتم به محیط‌های سه‌بعدی و کاربرد در سناریوهای چند ربات می‌تواند گام مهمی در تحقیقات آینده باشد و زمینه‌ی استفاده‌ی گسترده‌تر آن در سامانه‌های هوشمند فراهم سازد.

۷- مراجع

- [۳] Liu L, Wang X, Yang X, Liu H, Li J, Wang P. Path planning techniques for mobile robots: Review and prospect. *Expert Systems with Applications*. ۲۰۲۳;۲۲۷:۱۲۰۲۰۴.
- [۴] Lei Z, Wang W, Zhu Z, Ma J, Ge SS. Safe Motion Planning for Multi-Vehicle Autonomous Driving in Uncertain Environment. *IEEE Robotics and Automation Letters*. ۲۰۲۰;۱۰(۳):۲۱۹۹-۲۰۶.
- [۵] Hewawasam HS, Ibrahim MY, Appuhamillage GK. Past, Present and Future of Path-Planning Algorithms for Mobile Robot Navigation in Dynamic Environments. *IEEE Open Journal of the Industrial Electronics Society*. ۲۰۲۲;۳:۳۰۳-۶۰.
- [۶] Amores D, Tanin E, Vasardani M. Flexible Paths: A Path Planning Approach to Dynamic Navigation. *IEEE Transactions on Intelligent Transportation Systems*. ۲۰۲۴;۲۵(۶):۴۷۹۰-۸۰۸.
- [۷] Alarabi S, Santora M, editors. Review: Path Planning Techniques for Automated Guided Vehicles (AGVs). ۲۰۲۴ ۹th Asia-Pacific Conference on Intelligent Robot Systems (ACIRS); ۲۰۲۴ ۱۸-۲۰ July ۲۰۲۴.
- [۸] Lin Z, Wu K, Shen R, Yu X, Huang S. An Efficient and Accurate A-Star Algorithm for Autonomous Vehicle Path Planning. *IEEE Transactions on Vehicular Technology*. ۲۰۲۴;۷۳(۶):۹۰۰۳-۸.
- [۹] Yang C, Lin Z. Dictionary-Labeled A^* : Optimal Path-and-Posture Planning for Mobile Robots With Turning Radius Constraints. *IEEE Robotics and Automation Letters*. ۲۰۲۴;۹(۱۲):۱۰۷۴۰-۷.
- [۱۰] Vagale A, Bye RT, Oucheikh R, Osen OL, Fossen TI. Path planning and collision avoidance for autonomous surface vehicles II: a comparative study of algorithms. *Journal of Marine Science and Technology*. ۲۰۲۱;۲۶(۴):۱۳۰۷-۲۳.
- [۱۱] Karur K, Sharma N, Dharmatti C, Siegel JE. A Survey of Path Planning Algorithms for Mobile Robots. *Vehicles*. ۲۰۲۱;۳(۳):۴۴۸-۶۸.
- [۱۲] Liu H, Wu G, Zhou L, Pedrycz W, Suganthan PN. Tangent-Based Path Planning for UAV in a ۳-D Low Altitude Urban Environment. *IEEE Transactions on Intelligent Transportation Systems*. ۲۰۲۳;۲۴(۱۱):۱۲۰۶۲-۷۷.
- [۱۳] Zafar MM, Anjum ML, Hussain W. LTA*: Local tangent based A^* for optimal path planning. *Autonomous Robots*. ۲۰۲۱;۴۵(۲):۲۰۹-۲۷.
- [۱۴] Rashid A, Marhoon A, Yahia Z. An algorithm for Path planning with polygon obstacles avoidance based on the virtual circle tangents. *Iraq Journal Electrical and Electronic Engineering*. ۲۰۱۶;۱۲:۲۲۱.
- [۱۵] Zhang H, Xie X, Wei M, Wang X, Song D, Luo J, editors. An Improved Goal-bias RRT algorithm for Unmanned Aerial Vehicle Path Planning. ۲۰۲۴ IEEE
- [۱] Madridano Á, Al-Kaff A, Martín D, de la Escalera A. Trajectory planning for multi-robot systems: Methods and applications. *Expert Systems with Applications*. ۲۰۲۱;۱۷۳:۱۱۴۶۶۰.
- [۲] Aggarwal S, Kumar N. Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges. *Computer Communications*. ۲۰۲۰;۱۴۹:۲۷۰-۹۹.

International Conference on Mechatronics and Automation (ICMA); ۲۰۲۴ Aug. ۲۰۲۴.

[۱۶] Kim M, Ahn J, Park J. TargetTree-RRT*: Continuous-Curvature Path Planning Algorithm for Autonomous Parking in Complex Environments. IEEE Transactions on Automation Science and Engineering. ۲۰۲۴;۲۱(۱):۶۰-۶۷.

[۱۷] Tan CS, Mohd-Mokhtar R, Arshad MR. A Comprehensive Review of Coverage Path Planning in Robotics Using Classical and Heuristic Algorithms. IEEE Access. ۲۰۲۱;۹:۱۱۹۳۱۰-۴۲.

[۱۸] [Mahdavi F, Shamaghdari S. Optimal Formation Control for Unmanned Aerial Vehicle Teams with Collision Avoidance Constraint and Unknown Dynamics. Journal of Aerospace Mechanics. ۲۰۲۳;۱۹\(۱\):۶۱-۷۹. \(In Persian\)](#)

[۱۹] Rahman MA, Azad MAK. To Escape Local Minimum Problem for Multi-Agent Path Planning using Improved Artificial Potential Field-Based Regression Search Method. Proceedings of the ۲۰۱۷ International Conference on Information Technology; Singapore, Singapore: Association for Computing Machinery; ۲۰۱۷. p. ۳۷۱-۶.

[۲۰] Vasudevan A, Kumar A, Theetharapan N, Bhuvaneswari NS, editors. Robot Operating System Based Path Planning. Intelligent Data Communication Technologies and Internet of Things; ۲۰۲۰, ۲۰۲۰//; Cham: Springer International Publishing.

[۲۱] Yue Y, Zhang D, Niu L, Zuo X. OPTIMAL PATH PLANNING METHOD FOR UAV BASED ON THE TANGENT POINT ALGORITHM IN URBAN ENVIRONMENT. International Journal of Innovative Computing, Information and Control. ۱۹:۳۲.

[۲۲] [Zardashti R, Nikkhah AA, Yazdanpanah MJ. Considering Terrain Contour Matching System Requirements in the TF/TA Based Constraint Optimal Trajectory Design for UAVs. Aerospace Mechanics Journal. ۲۰۱۷;۱۲\(۴\):۲۵. \(In Persian\)](#)

[۲۳] [Radmanesh A, Radmanesh H, qorbani m, Mohamadhossein H. Six Degree of Freedom Simulation and Hybrid Guidance of Parafoil-Payload Systems Considering Wind and Terrain Constraints. Aerospace Mechanics. ۲۰۲۱;۱۷\(۱\):۲۱-۳۶. \(In Persian\)](#)

[۲۴] Boyd S, Vandenberghe L. Convex Optimization. Cambridge: Cambridge University Press; ۲۰۰۴.