



Sahand University
of Technology

DOR:

[20.1001.1.23223146.1404.12.1.3.6](https://doi.org/10.1001.1.23223146.1404.12.1.3.6)

Journal of Nonlinear
Systems in Electrical
Engineering

Vol.12, No.1

Spring and Summer 2025

ISSN: 2322 – 3146

<http://jnsee.sut.ac.ir>

Mobile Robot Localization in Indoor Environments Using a Hybrid Extended Kalman Filter and Rapidly-Exploring Random Tree Algorithm

Mohammadrasoul Haj Ali, Ramazan Havangi*

Faculty of Electrical and Computer Engineering, University of Birjand, Birjand, Iran

Email: havangi@birjand.ac.ir

Abstract

This paper presents an integrated framework for mobile robot navigation in indoor environments, based on the fusion of the Extended Kalman Filter (EKF) and the Rapidly-exploring Random Tree* (RRT) algorithm. The objective of this hybrid approach is to simultaneously enhance path planning accuracy and localization precision under conditions of sensor noise and dynamic, uncertain obstacles. Initially, the RRT algorithm generates an optimal path from the start to the goal point, considering existing obstacles. Subsequently, for high-accuracy path following, the robot's position is estimated in real-time using the EKF. In this method, the prediction step of the filter relies on the robot's kinematic model and odometry data, while the update step is performed using Lidar sensor measurements. The precise localization provided by the EKF enables the robot to follow the proposed RRT* path with minimal error. A key feature of this framework is the closed-loop interaction between localization and path planning; if the estimation error increases or unexpected environmental changes occur, the path planning process is reactivated by the RRT* module. This design increases the system's adaptability and stability in the face of environmental uncertainties. Simulation results demonstrate that this framework significantly improves localization accuracy, achieving a Root Mean Square Error (RMSE) of less than 0.055 and a 73% increase in precision compared to pure odometry. Furthermore, the paths generated by RRT* show considerable improvement in length, smoothness, and safety over those produced by the basic RRT algorithm.

Keywords

Extended Kalman Filter; Indoor Environments; Sampling-based Algorithms; Rapidly-exploring Random Tree; Mobile Robot Localization.

1. Short Introduction

Accurate localization of mobile robots in unknown or dynamic environments remains a fundamental challenge in autonomous navigation. While common methods like odometry suffer from cumulative error, and advanced filters like EKF improve estimation, they often operate independently of path planning. Similarly, path planners like RRT* may generate inefficient routes if the robot's positional uncertainty is high. This paper is motivated by the need to bridge this gap, proposing a novel closed-loop EKF-RRT* framework that synergistically integrates precise, real-time state estimation with reactive and optimal path planning to enhance overall navigation robustness in complex indoor settings.

2. Proposed Work

The proposed system establishes a closed-loop integration between an Extended Kalman Filter (EKF) for localization and a Rapidly-exploring Random Tree (RRT*) algorithm for path planning. The EKF fuses noisy wheel odometry data with Lidar measurements to provide a reliable real-time pose estimate and, crucially, a measure of estimation uncertainty (covariance). The RRT module uses this estimated position to generate an optimal, collision-free path. The core innovation is the feedback link: if the positional uncertainty from the EKF exceeds a threshold—indicating potential localization drift or new obstacles—the system triggers the RRT* to replan a new path from the robot's current estimated location. This ensures the path remains feasible despite uncertainties.

Simulation results in indoor environments validate the framework's superiority. The EKF-based localization achieved a Root Mean Square Error (RMSE) of 0.055, outperforming pure odometry where error grew unbounded. The closed-loop interaction produced paths that were not only collision-free but also 73% more accurate in tracking compared to odometry. The paths were shorter and smoother than those from a standard RRT, as frequent replanning from accurate poses led to more efficient tree growth and obstacle avoidance.

3. Conclusion

This paper successfully developed and validated a hybrid EKF-RRT* framework for robust mobile robot navigation. The closed-loop design synergizes high-precision localization with adaptive path planning, where each module informs and improves the other. Key results demonstrate a 73% increase in localization accuracy over odometry and the generation of safer, more optimal paths. The proposed system represents a significant step toward reliable autonomous navigation for service, industrial, and rescue applications in uncertain indoor environments.



دانشگاه صنعتی سهند

DOR:

20.1001.1.23223146.1404.12.1.3.6

نشریه سالانه علمی غیرخطی در مهندسی برق

دوره ۱۲ - شماره ۱

بهار و تابستان ۱۴۰۴

صفحات ۵۷ الی ۸۰

ISSN: 2322-3146

http://jnsee.sut.ac.ir

موقعیت یابی ربات متحرک در محیط های داخلی با استفاده از ترکیب

فیلتر کالمن توسعه یافته و الگوریتم مسیریابی درخت تصادفی

محمد رسول حاج علی، رمضان هاوونگی*

دانشکده مهندسی برق و کامپیوتر - دانشگاه بیرجند - بیرجند - ایران

نویسنده مسوول havangi@birjand.ac.ir

تاریخ پذیرش: ۱۴۰۴/۴/۲۴

تاریخ بازنگری: ۱۴۰۴/۳/۲۳

تاریخ ارسال: ۱۴۰۴/۱/۱۸

چکیده

واژه‌های کلیدی

فیلتر کالمن توسعه یافته،

محیط‌های داخلی،

الگوریتم‌های مبتنی بر نمونه برداری،

الگوریتم کاوش سریع درخت

تصادفی،

موقعیت‌یابی ربات متحرک.

در این مقاله، یک چارچوب یکپارچه برای ناوبری ربات متحرک در محیط‌های داخلی ارائه شده است که مبتنی بر ترکیب فیلتر کالمن توسعه یافته (EKF) و الگوریتم درخت تصادفی با کاوش سریع (RRT*) می‌باشد. هدف از این ترکیب، بهبود همزمان دقت مسیریابی و موقعیت‌یابی در شرایطی است که داده‌های حسگر دارای نویز هستند و محیط دارای موانع پویا و نامعین می‌باشد. در گام اول، RRT* مسیر بهینه‌ای از نقطه شروع تا هدف را با در نظر گرفتن موانع موجود ایجاد می‌کند. سپس، برای دنبال کردن دقیق مسیر، تخمین موقعیت ربات در طول حرکت با استفاده از EKF انجام می‌شود. در این روش، مرحله پیش‌بینی فیلتر مبتنی بر مدل حرکتی ربات و داده‌های ادومتری است و مرحله به‌روزرسانی از طریق اندازه‌گیری‌های حسگر لیدار صورت می‌گیرد. موقعیت‌یابی دقیق به‌دست آمده از EKF، به ربات امکان می‌دهد که مسیر پیشنهادی RRT* را با خطای حداقلی دنبال کند. نکته کلیدی در این چارچوب، ارتباط حلقه بسته بین موقعیت‌یابی و مسیریابی است؛ به طوری که در صورت افزایش خطای تخمین یا تغییرات غیرمنتظره در محیط، فرآیند برنامه‌ریزی مسیر مجدداً توسط RRT* فعال می‌شود. این طراحی باعث افزایش تطبیق‌پذیری و پایداری سیستم در مواجهه با عدم قطعیت‌های محیطی می‌گردد. نتایج حاصل از شبیه‌سازی‌های متعدد نشان می‌دهد که این چارچوب، توانسته است دقت موقعیت‌یابی را به میزان قابل توجهی بهبود دهد. میانگین خطای ریشه میانگین مربعات (RMSE) موقعیت‌یابی در آزمایش‌ها کمتر از ۰/۰۵۵ گزارش شده است. همچنین، در مقایسه با الگوریتم‌های پایه مانند ادومتری صرف، دقت موقعیت‌یابی به میزان ۷۳٪ افزایش یافته است. مسیرهای تولیدشده توسط RRT* نیز از نظر طول، همواری و ایمنی نسبت به مسیرهای ایجادشده توسط نسخه اولیه RRT* بهبود چشم‌گیری داشته‌اند. این نتایج نشان می‌دهد که ادغام EKF و RRT* نه تنها به بهبود عملکرد هر کدام از این ماژول‌ها منجر شده، بلکه به طراحی یک سیستم ناوبری هوشمند و مقاوم کمک کرده است که قابلیت استفاده در کاربردهای ربات‌های خدماتی، صنعتی و امدادی را داراست.

۱- مقدمه

موقعیت یابی دقیق ربات های متحرک، به ویژه در محیط های ناشناخته یا پویا، یکی از چالش های اساسی در حوزه ناوبری خودکار به شمار می رود. ربات های متحرک در دهه های اخیر دستاوردهای قابل توجهی در حوزه هایی مانند تولید صنعتی، علوم پزشکی، کشاورزی، آموزش و تحقیقات فضایی داشته اند. برای تحقق ناوبری ایمن و رسیدن به اهداف کنترلی، ربات باید از دقت بالایی در موقعیت یابی برخوردار باشد. در یک سیستم ناوبری معمول، ربات مراحل مختلفی شامل ادراک، موقعیت یابی، شناخت و کنترل حرکت را طی می کند. در مرحله ادراک، داده های سنسورها به اطلاعات معنادار تبدیل می شوند؛ در مرحله موقعیت یابی، ربات با استفاده از داده های حسگرهای داخلی و خارجی موقعیت خود را در فضای کاری تخمین می زند؛ در مرحله شناخت، مسیر حرکت برنامه ریزی می شود و در نهایت، با کنترل حرکت، خروجی های موتور تنظیم می شوند تا ربات به مقصد برسد. در این میان، مرحله موقعیت یابی از اهمیت ویژه ای برخوردار است و در سال های اخیر مورد توجه قابل توجهی از سوی پژوهشگران حوزه رباتیک و سیستم های هوشمند قرار گرفته است [۱]. یکی از روش های متداول در این زمینه، الگوریتم ادومتری است که در ربات های متحرک به کار می رود [۲-۴]. با این حال، خطای تجمعی ناشی از ادومتری با افزایش مسافت طی شده افزایش می یابد و منجر به عدم قطعیت در موقعیت یابی می شود. الگوریتم های فیلتر کالمن توسعه یافته (EKF) به واسطه ساختار بازگشتی خود می توانند با به روزرسانی های پیوسته خطا را کاهش داده و دقت ناوبری را بهبود بخشند [۵].

در سال های اخیر، تعداد قابل توجهی از مطالعات در حوزه قابلیت های ناوبری خودکار و پایداری ربات های متحرک منتشر شده است. برای مثال، در [۶] از رویکرد مبتنی بر فیلتر کالمن توسعه یافته (EKF) در مدلی که از حسگر سونار استفاده می کند برای موقعیت یابی ربات متحرک بهره گرفته شده است. همچنین در [۷]، رویکرد فیلترینگ مبتنی بر EKF به کار گرفته شده است که از نقشه به عنوان منبع کمکی برای موقعیت یابی ربات های متحرک استفاده می کند. در [۸]، مقایسه ای بین فیلتر کالمن خطی (KF) و EKF ارائه شده است که تحلیل شبیه سازی ها نشان می دهد EKF در تخمین موقعیت ربات متحرک به طور قابل توجهی بهتر عمل می کند. در این رابطه، لازم به ذکر است که فیلتر کالمن خطی که در سیستم هایی با دینامیک خطی کاربرد دارد، در سیستم های غیرخطی نیز با اعمال اصلاحاتی قابل استفاده است؛ به این صورت که در هر مرحله از تخمین حالت، با استفاده از سری تیلور، دینامیک سیستم غیرخطی خطی سازی می شود تا بتوان از همان فیلترهای کالمن خطی بهره گرفت [۹]. در این منبع همچنین نشان داده شده است که الگوریتم فیلتر کالمن خطی تا حدی قابلیت تطبیق با شرایط غیرخطی سیستم را دارد. باید توجه داشت که اغلب فیلترهای کالمن به صورت گسسته پیاده سازی می شوند.

در [۱۰]، یک تکنیک ناوبری بصری بدون رانش برای موقعیت یابی ربات متحرک ارائه شده است که در آن فناوری موقعیت یابی باند فوق وسیع (UWB) به عنوان منبع داده کمکی استفاده می شود. همچنین در [۱۱] یک روش موقعیت یابی با استفاده از EKF برای ترکیب داده های حاصل از دو حسگر مختلف، یعنی ادومتری و بینایی کامپیوتری پیاده سازی شده است. نتایج موقعیت یابی در محیط شبیه سازی شده با نقاط شروع شناخته شده و ناشناخته ارائه شده است؛ با این حال، مسئله برنامه ریزی مسیر در این مطالعه مورد بررسی قرار نگرفته است. در [۱۲] تمرکز بر مکان یابی دقیق مبتنی بر شبکه حسگر بی سیم (WSN) برای ربات متحرک است که در

آن با استفاده از شناسایی شرایط عدم دید مستقیم (NLOS) تلاش شده خطاهای بزرگ تر ناشی از روش های سنتی کاهش یابد. به منظور حل این مشکل، توزیع موقعیت محاسبه شده از چند اندازه گیری مختلف برای تخمین مکان ربات متحرک استفاده شده است. در [۱۳] به بررسی موقعیت یابی ربات متحرک بر اساس الگوریتم EKF و یک نقشه مبتنی بر ویژگی های زوایا پرداخته شده است. در این مقاله، زوایای گوشه به عنوان ویژگی های محیط شناسایی و فرآیند استخراج دقیق این ویژگی ها شرح داده شده است. همچنین مدل حرکتی ربات و داده های ادومتری توضیح داده شده و الگوریتم EKF به طور کامل ارائه شده است. در [۱۴]، موقعیت یابی یک ربات رهبر و یک ربات دنبال کننده بر اساس ترکیب داده های حسگرهای مختلف ارائه شده است. در این مطالعه، ربات رهبر مجهز به یک حسگر در بالای خود است که نشانگرهایی با چهار شناسه مختلف بر روی آن نصب شده اند. ربات دنبال کننده دارای پلتفرم چرخش با یک درجه آزادی است که حسگر اینتل روی آن نصب شده است. داده های رمزگذار چرخ و حسگرهای IMU در هر دو ربات با استفاده از EKF ترکیب می شوند تا وضعیت ربات ها به دست آید. کنترل کننده PID حسگر اینتل روی ربات دنبال کننده را می چرخاند تا نشانگرهای ربات رهبر را دنبال کند و موقعیت ربات رهبر پس از شناسایی نشانگرها محاسبه می شود. از طرفی، در زمینه برنامه ریزی مسیر، در [۱۵] الگوریتم RRT^* برای جمع آوری واگن برقی چمدان رباتیک در فرودگاه ها به کار رفته است. این مقاله نشان می دهد که الگوریتم های برنامه ریزی حرکت مبتنی بر ریسک موجود به طور موثری خطرات برخورد با موانع استاتیک و پویا را کاهش می دهند. الگوریتم RRT^* برخلاف الگوریتم های سنتی که فقط به یک درخت ریشه دار یا درخت های دوتایی تکیه دارند، چندین درخت فرعی مستقل را شامل می شود که هر کدام به صورت جداگانه محیط اطراف خود را بررسی می کنند. در همان زمان، درخت ریشه دار اولیه اطلاعات اکتشافی را از این درخت های فرعی جمع آوری و پیشرفت سریع به سمت وضعیت هدف را تسهیل می کند. ارزیابی های انجام شده، شامل شبیه سازی ها و مطالعات محیطی در دنیای واقعی، نشان می دهد که RRT^* از نظر کارایی و استحکام در برنامه ریزی مسیر بهتر از الگوریتم های مبتنی بر ریسک یک طرفه و دوطرفه عمل می کند.

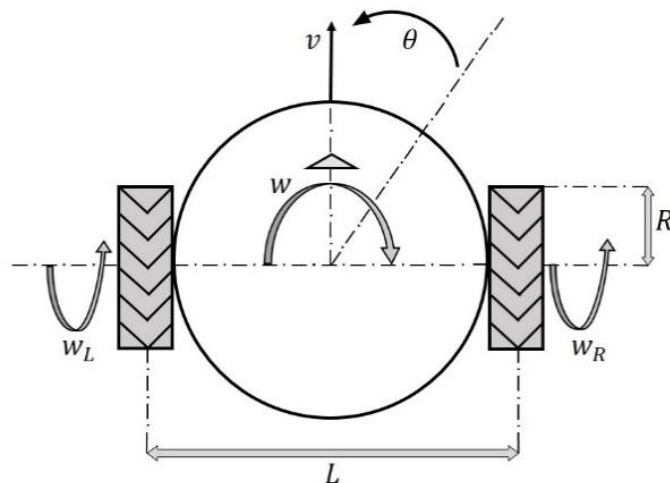
در [۱۶]، روش جدیدی برای برنامه ریزی مسیر ارائه شده است که عدم قطعیت حرکت در محیط های طبیعی را در نظر می گیرد. در این روش، ذرات برای انتشار عدم قطعیت در محیط های پیچیده و با انواع زمین مختلف استفاده می شوند. الگوریتم RRT^* بر اساس میزان عدم قطعیت هر گره به منظور جلوگیری از انباشت افزایش عدم قطعیت موقعیت گسترش یافته است. در نتیجه، مسیر تولید شده زمان پیگیری مسیر و برنامه ریزی مجدد را بر اساس اطلاعات موقعیت یابی نادرست کاهش می دهد. نتایج نشان می دهد که روش پیشنهادی عدم قطعیت موقعیت را کاهش داده و همزمان احتمال اجتناب از برخورد و رسیدن به منطقه هدف را در مقایسه با روش های مرسوم حفظ می کند.

در [۱۷] به محدودیت های الگوریتم RRT سنتی در برنامه ریزی مسیر بازوی رباتیک پرداخته شده است، از جمله عملکرد تصادفی، سرعت برنامه ریزی آهسته، مسیرهای غیرهموار و گوشه های بیش از حد. در این مطالعه، الگوریتم RRT بهبود یافته ای با استراتژی سوگیری هدف و رویکرد میدان پتانسیل مصنوعی پیشنهاد شده است. از نظر کاربردهای ناوبری هوایی، در [۱۸] فیلتر کالمن مقاوم با محدودیت حرکتی برای مکان یابی ربات در محیط هوایی که با انسداد سیگنال رادیویی و ویژگی های بصری مشخص می شود، توسعه یافته است. این مقاله عملکرد بهینه EKF را در کاربردهای ناوبری هوایی نشان می دهد. علاوه بر این، در [۱۹] الگوریتم ES-EKF برای موقعیت یابی بلادرننگ در محیط های فاقد GNSS پیشنهاد شده است که سیستم پیشنهادی را در برابر خرابی سنسورهای کمکی

مقاوم تر می سازد. موقعیت یابی مبتنی بر فیلتر کالمن به شیوه ای کارآمد و دقیق مسئله ردیابی موقعیت ربات را حل می کند. این رویکرد ترکیبی حسگر بهینه به ربات امکان می دهد از مکان شناخته شده اولیه، موقعیت خود را به صورت پیوسته ردیابی کند [۲۰].

با توجه به مرور مطالعات پیشین، چالش های اصلی در ناوبری ربات های متحرک شامل دقت پایین در موقعیت یابی به دلیل نویز حسگرها، عدم قطعیت های محیطی و مسیریابی با کیفیت پایین تولید شده توسط الگوریتم های مسیریابی است. هدف اصلی این مقاله ارائه یک چارچوب ترکیبی مبتنی بر ادغام الگوریتم درخت تصادفی با کاوش سریع (RRT^*) و فیلتر کالمن توسعه یافته (EKF) است که بتواند به صورت یکپارچه به بهبود دقت ناوبری، کاهش خطای موقعیت یابی و افزایش تطبیق پذیری مسیر در شرایط واقعی پردازد. نوآوری مقاله در طراحی یک ساختار حلقه بسته میان موقعیت یابی و مسیریابی است که امکان بازبرنامهریزی مسیر در صورت افزایش خطای تخمین را فراهم می سازد. این چارچوب منجر به تولید مسیریابی هموار، ایمن و قابل اجرا می شود و دقت مسیریابی را نیز بهبود می بخشد. روش پیشنهادی قابلیت کاربرد در ربات های متحرک در حوزه های صنعتی، خدماتی و امدادی را دارد.

ساختار بقیه مقاله به شرح زیر است: در بخش ۲، مسئله موقعیت یابی ربات و روابط مرتبط به طور جامع توضیح داده شده است. در بخش ۳، ترکیب فیلتر کالمن توسعه یافته (EKF) و الگوریتم برنامه ریزی مسیر RRT^* به طور کامل بررسی شده است. در بخش ۴، نتایج پژوهش ارائه شده و در نهایت، بخش ۵ شامل نتیجه گیری کلی و دستاوردهای مقاله می باشد.



شکل ۱- مدل سینماتیک ربات متحرک.

۲- مسئله موقعیت یابی

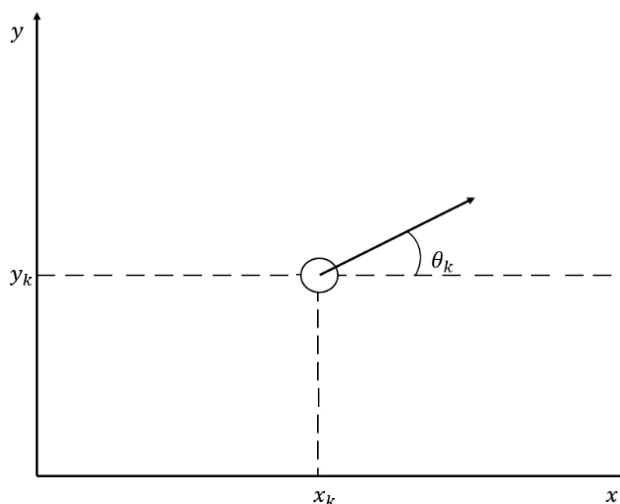
یک ربات متحرک در حین ناوبری در یک محیط شناخته شده با استفاده از ادومتر موقعیت خود را ردیابی می کند. با این حال، عدم قطعیت ادومتری باعث می شود که ربات در مورد صحت موقعیت فعلی خود با عدم قطعیت مواجه باشد. بنابراین، ربات باید خود را نسبت به نقشه محیط موقعیت یابی کند. این امر باعث می شود که عدم قطعیت موقعیت ربات افزایش نیابد. برای مکان یابی، ربات از حسگرهای خارجی خود مانند حسگر لیدار^۱ و حسگر اولتراسونیک^۲ استفاده می کند تا اطلاعاتی درباره محیط خود دریافت کند.

¹ Lidar

² Ultrasonic sensor

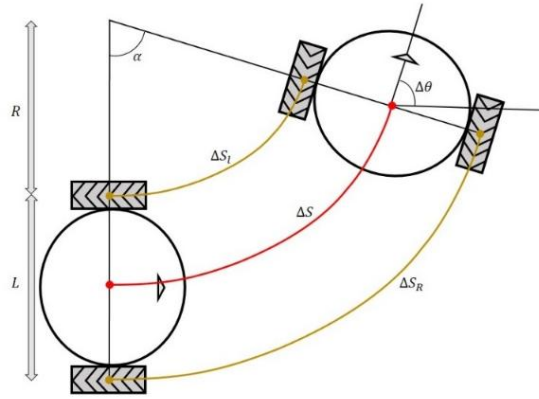
اطلاعات حسگرها را می توان با داده های ادومتری ربات ترکیب کرد تا تعیین موقعیت ربات ممکن شود. اساساً، حتی با وجود حسگر موقعیت یاب جهانی (GPS^1)، موقعیت دقیق ربات نمی تواند به طور مستقیم اندازه گیری شود. سینماتیک یک ربات به پیکربندی چرخ های آن و نوع آن ها بستگی دارد. ساده ترین آن یک پیکربندی دیفرانسیل است. این وسیله نقلیه دارای دو چرخ مستقل است که می توان از آن ها برای کنترل سرعت جلو و زاویه استفاده کرد. در شکل (۱) مدل سینماتیک ربات متحرک نمایش داده شده است. از نظر تئوری، می توان موقعیت واقعی ربات را تا زمانی محاسبه کرد که ساختار ربات به خوبی شناخته شده باشد. سرعت چرخ ربات را می توان به دقت اندازه گیری کرد. از این رو، می توان خطاهای ربات را در طول سفر تشخیص داد و مسیر را بر اساس آن تنظیم نمود. این کار از طریق سینماتیک انجام می شود. ربات باید توانایی تشخیص تغییرات حرکتی هر چرخ را داشته باشد. تخمین موقعیت یک ربات متحرک دشوار است، هیچ راه مستقیمی برای اندازه گیری از حسگرها وجود ندارد و باید حرکت ربات در طول زمان یکپارچه شود همچنین عدم دقت تخمین حرکت (به عنوان مثال، لغزش چرخ) مشکل ساز است و باید یک مدل ربات بر اساس آن تعریف شود. هندسه هر چرخ به حرکت کمک می کند و همچنین محدودیت هایی را اعمال می کند؛ به عنوان مثال، لغزش جانبی در چرخ ها خود به مرور باعث منحرف شدن ربات و در نهایت ایجاد عدم قطعیت در موقعیت یابی می شود. یک ربات متحرک دیفرانسیل را در نظر بگیرید. شکل (۲) نشان دهنده موقعیت ربات با توجه به سیستم مختصات جهانی است. موقعیت ربات با بردار (۱) نشان داده می شود.

$$X_k = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} \quad (1)$$



شکل ۲- موقعیت ربات.

¹ Global Positioning System



شکل ۳- یک نمونه چرخش ساده که بیانگر پارامترهای اساسی سینماتیک ربات است.

با توجه به شکل (۱)، حرکت ربات با دو پارامتر اصلی سنجیده می شود: سرعت خطی پیشروی v و سرعت زاویه ای چرخش w . شعاع چرخ های ربات و فاصله بین دو چرخ به ترتیب با R و L نمایش داده می شوند. همچنین زاویه θ بیانگر تغییرات زاویه ای وضعیت ربات نسبت به محور مرجع است. با استناد به شکل (۳)، مسافت طی شده توسط چرخ چپ ΔS_L و چرخ راست ΔS_R داده می شود. همچنین کل مسافت طی شده توسط ربات با ΔS نشان داده شده است. زاویه α ، زاویه بین میله متصل کننده دو چرخ و محور افقی را نشان می دهد. روابط مربوط به محاسبه ΔS_L و ΔS_R و همچنین ΔS به صورت زیر است:

$$\Delta S_L = R\alpha \quad (2)$$

$$\Delta S_R = (R + L)\alpha \quad (3)$$

$$\Delta S = \frac{\Delta S_R + \Delta S_L}{2} \quad (4)$$

با توجه به شکل (۳) مسافت پیموده شده توسط چرخ سمت چپ و چرخ سمت راست قابل محاسبه است. در مرحله بعد زاویه جدید ربات با رابطه (۵) محاسبه می شود:

$$\Delta\theta = \frac{\Delta S_R - \Delta S_L}{L} \quad (5)$$

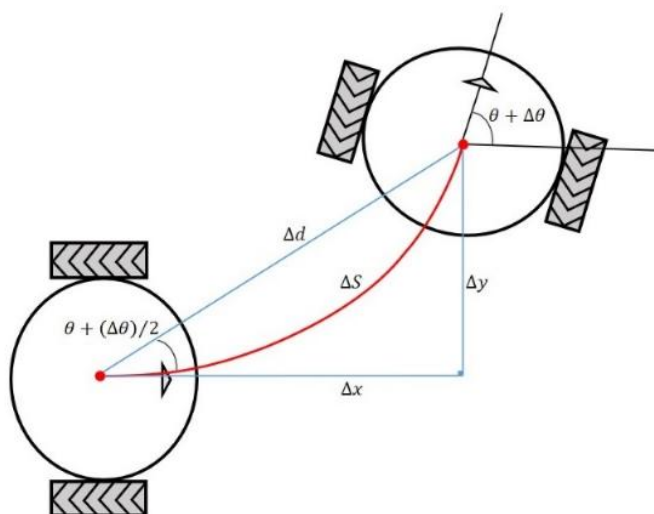
از آنجاییکه از قبل نحوه محاسبه زاویه و مسافت طی شده توسط چرخ های ربات در دسترس است پس امکان محاسبه موقعیت ربات وجود دارد. طبق شکل (۴) بدست آوردن معادلات مربوطه ممکن است.

موقعیت فعلی یک درایو دیفرانسیل را می توان از اطلاعات موقعیت اولیه آن با اضافه کردن فاصله های افزایشی سفر $(\Delta x, \Delta y, \Delta\theta)$ تخمین زد و در نهایت می توان فاصله سنجی ادومتری را به صورت زیر محاسبه کرد:

$$\Delta x = \Delta s \cos\left(\theta + \frac{\Delta\theta}{2}\right) \quad (6)$$

$$\Delta y = \Delta s \sin \left(\theta + \frac{\Delta \theta}{2} \right) \quad (7)$$

وضعیت واقعی ربات تابعی از وضعیت قبلی آن است. در معادله (۸)، فرض بر این است که بردار X_k بیانگر وضعیت فعلی ربات در زمان k است. این وضعیت با استفاده از بردار X_{k-1} ، که نمایانگر وضعیت قبلی ربات در لحظه $k-1$ است، محاسبه می شود. بر این اساس، معادله (۸) نحوه ی گذار از وضعیت پیشین به وضعیت کنونی را توصیف می کند. نکته حائز اهمیت آن است که سیستم مورد بررسی یک سیستم زمان گسسته است، به این معنا که وضعیت ها در گام های زمانی مجزا محاسبه می شوند و مدل سیستم بر مبنای همین گام های گسسته طراحی شده است.



شکل ۴- پارامترهای سینماتیک که مسافت پیموده شده در مسیر انحرافی با آن ها محاسبه می شود.

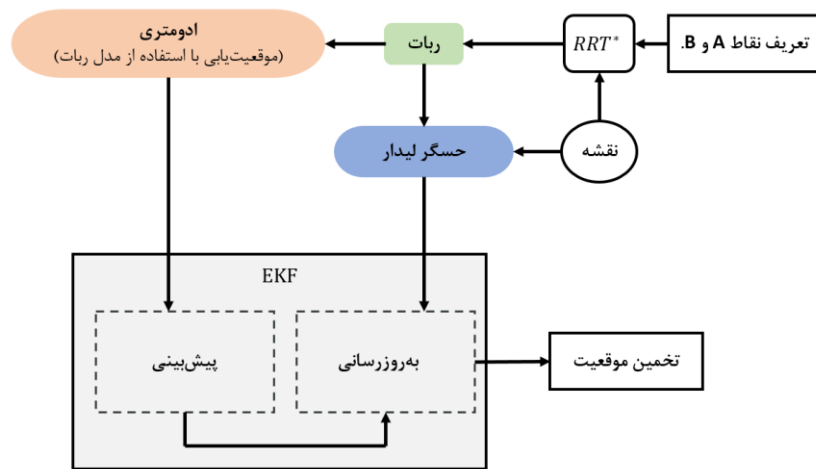
$$X_k = X_{k-1} + \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta \theta \end{bmatrix} = X_{k-1} + \begin{bmatrix} \Delta s \cos \left(\theta + \frac{\Delta \theta}{2} \right) \\ \Delta s \sin \left(\theta + \frac{\Delta \theta}{2} \right) \\ \Delta \theta \end{bmatrix} \quad (8)$$

از جمله مهم ترین عوامل ضعف روش ادومتری می توان به انباشته شدن خطاها و بسندگی نداشتن مدل حرکتی فرضی اشاره کرد. ادومتری بر پایه ی اندازه گیری چرخش چرخ ها و محاسبه ی مسافت پیموده شده ی ربات بر اساس این داده ها عمل می کند. با این حال، در طول زمان، خطاهای تصادفی و سینماتیک می توانند در اندازه گیری ها رخ دهند و به صورت تجمعی افزایش یابند، که این امر به تخمین های نادرست از موقعیت و جهت گیری ربات منجر می شود. علاوه بر این، روش ادومتری معمولاً یک مدل حرکت ساده شده را فرض می کند که در آن ربات یا به صورت خطی یکنواخت حرکت می کند یا چرخشی دقیق دارد. در حالی که در عمل، حرکات ربات های متحرک اغلب از رفتارهای غیرخطی محیط تأثیر می پذیرد. این اختلاف میان مدل ایده آل و واقعیت عملی، منشأ بروز خطاهایی در موقعیت یابی بر اساس ادومتری است. بنابراین، برای رفع یا کاهش اثرات این خطاها، بهره گیری از یک رویکرد همجوشی

حسگر نظیر فیلتر کالمن توسعه یافته توصیه می شود تا با ترکیب داده های ادومتری با اطلاعات سایر حسگرها مانند GPS یا لیدار، تخمینی دقیق تر از وضعیت ربات حاصل شود.

۳- موقعیت یابی ربات با EKF و الگوریتم RRT*

چارچوب کلی سیستم یکپارچه EKF-RRT* در شکل (۵) نشان داده شده است. فرآیند با تعریف نقاط A و B به عنوان نقطه شروع و هدف آغاز می شود و الگوریتم RRT* بهترین مسیر ممکن را برای ربات محاسبه می کند. حسگرهای داخلی ربات، شامل شمارنده های چرخ های چپ و راست، در کنار مدل حرکتی ربات برای تخمین اولیه موقعیت به کار می روند. همچنین، حسگر لیدار به عنوان یک حسگر خارجی نقشه محور، نقش کلیدی در اصلاح موقعیت نسبی ربات در مرحله به روزرسانی دارد. در این ساختار، از فیلتر کالمن توسعه یافته (EKF) در هر دو مرحله پیش بینی و تصحیح استفاده می شود؛ به طوری که مرحله پیش بینی مبتنی بر داده های ادومتری و مرحله تصحیح بر اساس اندازه گیری های لیدار انجام می گیرد. نکته مهم آن است که دقت موقعیت یابی ربات که توسط EKF تأمین می شود، نقش اساسی در امکان پذیر بودن و پایداری مسیر پیشنهادی RRT* دارد. مسیر بهینه محاسبه شده توسط RRT* بدون تخمین صحیح موقعیت ربات ممکن است در عمل غیرقابل پیاده سازی باشد. از سوی دیگر، هرگونه انحراف یا افزایش عدم قطعیت که توسط EKF شناسایی شود، می تواند منجر به فراخوانی مجدد RRT* و بازبرنامه ریزی مسیر شود. این تعامل دوطرفه، منجر به بهبود عملکرد ناوبری در شرایط غیرایستا و نامطمئن می شود.



شکل ۵- چهارچوب کلی رویکرد پیشنهادی.

۳-۱- موقعیت یابی ربات مبتنی بر EKF

EKF یک نسخه غیرخطی از فیلتر کالمن است که برای تخمین وضعیت یک سیستم دینامیکی غیرخطی با نویز گاوسی طراحی شده است. EKF از مدل ادومتری برای تخمین موقعیت ربات استفاده می کند. همچنین به کمک این رویکرد می توان اطلاعات حسگرهای داخلی و خارجی را ترکیب کرد. فرض کنید که موقعیت ربات متحرک در زمان $k - 1$ به صورت زیر است:

$$X_{k-1} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \end{bmatrix} \quad (9)$$

در لحظه k ، ربات به سمت موقعیت جدید X_k حرکت می کند. با استفاده از معادله (۱۰)، موقعیت جدید X_k در زمان k می تواند از موقعیت قبلی X_{k-1} پیش بینی شود:

$$X_k = f(X_{k-1}, u_k) = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \end{bmatrix} + \begin{bmatrix} \frac{\Delta s_r + \Delta s_l}{2} \cos\left(\theta_{k-1} + \frac{\Delta s_r - \Delta s_l}{2L}\right) \\ \frac{\Delta s_r + \Delta s_l}{2} \sin\left(\theta_{k-1} + \frac{\Delta s_r - \Delta s_l}{2L}\right) \\ \frac{\Delta s_r - \Delta s_l}{L} \end{bmatrix} \quad (10)$$

در رابطه (۱۰)، k زمان نمونه برداری و Δs_r جابجایی چرخ راست و Δs_l جابجایی چرخ چپ است. همچنین x_{k-1} و y_{k-1} موقعیت ربات در زمان $k-1$ و θ_{k-1} جهت ربات در زمان $k-1$ است. L طول محور ربات که دو چرخ چپ و راست را به هم متصل می کند، می باشد. اجزای ورودی کنترل u_k مطابق (۱۱) است:

$$u_k = \begin{bmatrix} \Delta s_l \\ \Delta s_r \end{bmatrix} \quad (11)$$

EKF به کمک بسط تیلور، دینامیک سیستم و مدل های اندازه گیری را خطی سازی می کند تا بتواند تخمین وضعیت و کوواریانس را پیش بینی و به روزرسانی نماید. در این رویکرد، دینامیک سیستم غیر خطی به صورت زیر مدل می شود:

$$X_k = f(X_{k-1}, u_{k-1}) + w_{k-1} \quad (12)$$

$$Z_k = h(X_k) + v_k \quad (13)$$

X_k بردار متغیرهای حالت در نمونه k است، همچنین w_k و v_k به ترتیب نویز فرآیند و نویز اندازه گیری در لحظه k هستند و Z_k خروجی اندازه گیری در همان لحظه است. با توجه به رابطه (۱۰)، تابع غیر خطی f را می توان به صورت رابطه (۱۴) بیان کرد:

$$f(X_{k-1}, u_{k-1}) = \begin{bmatrix} x_{k-1} + \Delta s \cos(\Delta\theta) \\ y_{k-1} + \Delta s \sin(\Delta\theta) \\ \theta_{k-1} + \Delta\theta \end{bmatrix} \quad (14)$$

رابطه (۱۵) ژاکوبین تابع غیر خطی سیستم است:

$$F = \begin{bmatrix} 1 & 0 & -\Delta s \sin(\theta_k + \Delta\theta) \\ 0 & 1 & \Delta s \cos(\theta_k + \Delta\theta) \\ 0 & 0 & 1 \end{bmatrix} \quad (15)$$

h تابع غیر خطی اندازه گیری سیستم و u_k سیگنال ورودی سیستم در نمونه k است. واضح است که تابع f با مدل واقعی و عملی سیستم اختلاف دارد. این اختلاف در نویز فرآیند مدل سازی خواهد شد، که ناشی از عدم قطعیت در مدل سازی ریاضی سیستم است. نویز اندازه گیری همان نویز سنسورهاست که برخی از متغیرهای حالت و یا تابعی از آن ها را اندازه گیری می کند.

نویز اندازه گیری و نویز فرآیند اغلب به صورت گاوسی مدل سازی می شوند و ماتریس کوواریانس آن ها به ترتیب با Q_k و R_k نمایش داده می شوند. با انجام تحلیل های آماری بر روی سنسورها می توان R_k را به دقت تعیین کرد؛ اما به دلیل ماهیت پیچیده و متنوع سیستم های غیر خطی، روش مشخص و قطعی برای تعیین R_k و Q_k وجود ندارد و این ماتریس ها معمولاً به صورت سعی و خطا تنظیم می شوند. الگوریتم EKF در دو فاز پیش بینی و به روز رسانی اجرا می شود. فرض کنید که تمام اطلاعات سیستم تا نمونه $k - 1$ در دسترس باشد. در این صورت، بر اساس مدل ریاضی سیستم و ورودی های موجود تا لحظه $k - 1$ ، تخمینی اولیه از متغیرهای حالت در لحظه k محاسبه می شود. این مرحله، فاز پیش بینی EKF نام دارد. در لحظه k ، اندازه گیری جدیدی توسط سنسورها انجام شده و در دسترس قرار می گیرد. در مرحله دوم، یعنی فاز به روز رسانی، تخمین اولیه بر اساس داده های جدید اصلاح می شود تا مقدار نهایی متغیرهای حالت در زمان k به دست آید. فاز پیش بینی توسط دو رابطه (۱۶) و (۱۷) انجام می شود:

$$\hat{X}_{k|k-1} = f(\hat{X}_{k-1}, u_{k-1}) \quad (16)$$

$$P_{k|k-1} = F_{k-1}P_{k-1}F_{k-1}^T + Q_{k-1} \quad (17)$$

در روابط بالا \hat{X}_{k-1} بردار حالت های تخمین در نمونه $k - 1$ و P_{k-1} ماتریس کواریانس در نمونه $k - 1$ و $\hat{X}_{k|k-1}$ پیش بینی متغیر حالت در نمونه k بر اساس اطلاعات سیستم تا نمونه $k - 1$ و $P_{k|k-1}$ نیز پیش بینی ماتریس کواریانس در لحظه k است. ماتریس F_k دینامیک حالت خطی شده است که از طریق خطی سازی تیلور به صورت زیر به دست می آید:

$$F_k = \left. \frac{\partial f}{\partial X} \right|_{\hat{X}_{k-1}} \quad (18)$$

H_k نیز دینامیک اندازه گیری خطی شده است که در فاز به روز رسانی استفاده می شود. این ماتریس نیز به صورت زیر محاسبه می شود:

$$H_k = \left. \frac{\partial h}{\partial X} \right|_{\hat{X}_{k-1}} \quad (19)$$

فاز به روز رسانی با استفاده از روابطی که در ادامه ارائه می شوند انجام می گیرد. این مرحله، از اطلاعات حسگر برای اصلاح و بهبود تخمین حالت به دست آمده در فاز پیش بینی بهره می برد. در این مقاله، سنسور یک آشکارساز لیدار است، نشانه ها در محیط عملیاتی مرتب شده اند. ربات از قبل موقعیت هر یک را می داند، هر نشانه یک شناسه منحصر به فرد دارد. بردار اندازه گیری سنسور آشکارساز شامل سه پارامتر r و α و \hat{t} است که هر کدام به ترتیب بیانگر فاصله از ربات تا لیدار، زاویه از ربات نسبت به لیدار و شناسه منحصر به فرد مربوط به نشانه هستند. مشاهدات حسگر با Z نشان داده می شود:

$$Z_k = \begin{bmatrix} r_k \\ \alpha_k \end{bmatrix} \quad (20)$$

خوانش حسگر لیدار، همراه با نویز اندازه گیری v با کواریانس R است. خوانش های مورد انتظار حسگر با \hat{Z}_k نمایش داده می شود و از طریق رابطه (۲۱) قابل محاسبه است:

$$\hat{Z}_k = h(\hat{X}_{k|k-1}, S_i) \quad (21)$$

در رابطه (۲۱)، S_i یک بردار حاوی مختصات نقاط عطف و شناسه مخصوص آن ها است که این مختصات از قبل تعریف شده اند:

$$S_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix} \quad (22)$$

در بردار (۲۲)، x_i و y_i مختصات مربوط به نشانه و \hat{l} شناسه مخصوص آن نشانه است. همچنین در (۱۷)، $\hat{X}_{k|k-1}$ بردار حالت بدست آمده از مرحله پیش بینی است. h تابعی است که خوانش مورد انتظار حسگر را با توجه به بردار حالت و بردار لیدار محاسبه می کند. در (۲۳)، تابع غیرخطی h آمده است:

$$h(\hat{X}_{k|k-1}, S_i) = \begin{bmatrix} l \\ \beta \end{bmatrix} = \begin{bmatrix} \sqrt{(S_{i,x} - x)^2 + (S_{i,y} - y)^2} \\ \text{atan2}(S_{i,x} - x, S_{i,y} - y) - \theta \end{bmatrix} \quad (23)$$

در (۲۴) ژاکوبین تابع غیرخطی h است:

$$H = \begin{bmatrix} \frac{-(S_{i,x} - x)}{\sqrt{(S_{i,x} - x)^2 + (S_{i,y} - y)^2}} & \frac{-(S_{i,y} - y)}{\sqrt{(S_{i,x} - x)^2 + (S_{i,y} - y)^2}} & 0 \\ \frac{(S_{i,y} - y)}{\sqrt{(S_{i,x} - x)^2 + (S_{i,y} - y)^2}} & \frac{-(S_{i,x} - x)}{\sqrt{(S_{i,x} - x)^2 + (S_{i,y} - y)^2}} & -1 \end{bmatrix} \quad (24)$$

در رابطه (۲۳)، با توجه به اینکه ربات مجهز به فاصله یاب لیزری است، این حسگر فاصله سمت l و زاویه β را اندازه گیری می کند. \hat{l} شناسه مخصوص مربوط به نشانه است. در رابطه (۲۴)، تابع اندازه گیری خطی سازی شده است. با رسیدن اندازه گیری های حسگر، فاز به روزرسانی اجرا می شود:

$$K = P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R_k)^{-1} \quad (25)$$

$$\hat{X}_k = \hat{X}_{k|k-1} + K_k (Z_k - \hat{Z}_k) \quad (26)$$

$$P_k = (I - K_k H_k) P_{k|k-1} \quad (27)$$

در فرمول (۱۹)، Z_k همان خروجی اندازه گیری شده توسط سنسورها است. بنابراین، در قدم اول بهره کالمن (K_k) محاسبه می شود. بهره کالمن مقداری است که میزان ادغام اندازه گیری در حالت جدید را مشخص می کند. سپس تخمین نهایی متغیرهای حالت \hat{X}_k و ماتریس کواریانس P_k در نمونه k محاسبه خواهد شد. اگر به عبارت به روزرسانی حالت ها دقت کنیم، متوجه خواهیم شد که در فاز به روزرسانی به حالت پیش بینی شده در گام اول یک ترم جدید اضافه شده است که متناسب با خطای پیش بینی است. عبارت $(Z_k - \hat{Z}_k)$ دقیقاً بیانگر اختلاف بین خروجی واقعی اندازه گیری شده توسط سنسورها و خروجی به دست آمده در فاز پیش بینی است.

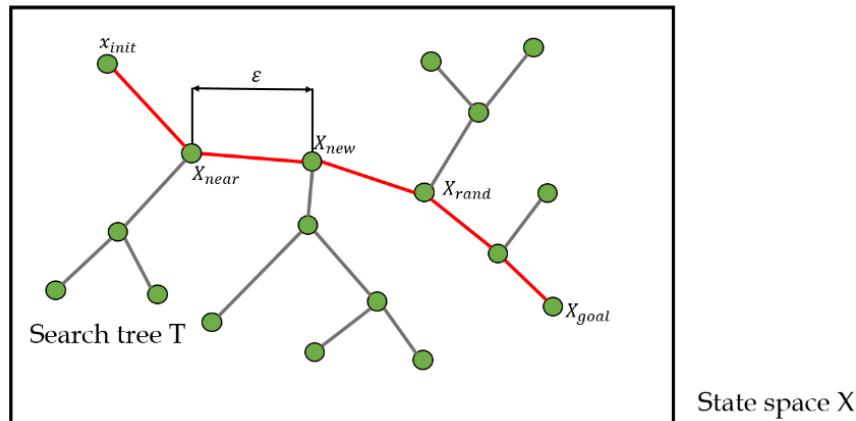
۳-۲- الگوریتم مسیریابی RRT*

الگوریتم RRT* یک روش برنامه ریزی مسیر مبتنی بر نمونه برداری است که با هدف یافتن مسیرهای مؤثر و امکان پذیر برای سیستم های رباتیکی در محیط های پیچیده توسعه یافته است [۲۲]. این الگوریتم با رشد تدریجی یک ساختار درختی از یک پیکربندی اولیه به سمت پیکربندی هدف، فضای حالت را به صورت هوشمندانه برای یافتن مسیرهای بهینه کاوش می کند [۲۳]. یکی از ویژگی های کلیدی RRT*، قابلیت بهینه سازی پویا ساختار درختی از طریق باز تنظیم گره ها است که موجب بهبود کیفیت مسیر و همگرایی آن به مسیر بهینه می شود [۲۴]. این ویژگی تطبیقی به RRT* اجازه می دهد تا فضاهای با ابعاد بالا و سیستم های دارای محدودیت های غیر هولونومیک را به طور مؤثر مدیریت کند [۲۵]. برای شروع، برخی از تعاریف و رویه های اولیه RRT* شرح داده شده است. پیکربندی ناحیه ناشناس به صورت X نشان داده می شود، جایی که X یک مجموعه با اعضای $X = \{x, y, \theta\}$ است. در اینجا x و y مختصات دو بعدی را نشان می دهند، در حالی که θ نشان دهنده جهت است. علاوه بر این، تمام مناطق مربوط به موانع را در بر می گیرد، در حالی که X_{free} مربوط به قسمت هایی از محیط است که بدون مانع هستند. رابطه زیر بیانگر ارتباط ریاضی بین این دو مجموعه است:

$$X_{free} = \frac{X}{X_{obs}} \quad (28)$$

در ادامه مفاهیم مهم تری برای بررسی مطرح می شوند این مفاهیم عبارت اند از: x_{init} و x_{goal} که به ترتیب به معنی نقطه شروع و نقطه هدف هستند [۲۶]. راه حل مسئله برنامه ریزی حرکت مسیر $\sigma: [0,1] \rightarrow X_{free}$ که $\sigma(0) = x_{init}$ و $\sigma(1) = x_{goal}$ است. در RRT*، همچنین یک الگوی درختی $T = (V, E)$ ساخته می شود که از مجموعه رئوس $V \subset X$ و شاخه های $E \subset V \times V$ تشکیل شده است. نمونه به طور تصادفی یک حالت $x_{rand} \in X_{free}$ را از فضای بدون مانع نمونه برداری می کند. فاصله با دو حالت $x, x' \in X$ ، تابع $dist(x, x')$ هزینه مسیر اتصال x و x' را برمی گرداند. تابع هزینه استفاده شده در این مقاله بر حسب فاصله اقلیدسی است. $Nearest$ و $Near$ با یک مجموعه $V \subset X$ و حالت $x \in X$ تعریف می شوند. این دو رویه نزدیکترین گره در درخت و تمام گره های نزدیک را که درون یک توپ با حجم $\gamma ((\log n)/n)^{1/d}$ در مرکز قرار دارند، برمی گرداند. در حدود x از نظر فاصله γ ثابتی است که در [۲۶] توضیح داده شده است، d بعد فضای X و n تعداد رئوس است. هدایت تابع $Steer(x, x')$ یک مسیر $\sigma: [0,1] \rightarrow X$ را برمی گرداند که x و x' را به هم متصل می کند. بررسی برخورد با یک مسیر $\sigma: [0,1] \rightarrow X$ تابع $InCollision(\sigma)$ بررسی می کند که آیا مسیر در X_{free} قرار دارد یا نه، و اگر نه، اولین حالت

شکست خورده x_{fail} را برمی گرداند. تابع هزینه با یک راس x ، اجازه می دهیم $C(x)$ هزینه کل مسیر منحصر به فرد باشد که از راس ریشه x_{init} شروع می شود، در امتداد گره های درخت می رود و به حالت x می رسد، در حالی که $C(\sigma)$ تابع هزینه از مسیر $\sigma: [0,1] \rightarrow X$ است. مرتب سازی لیست L ، مجموعه ای مرتب از عناصر با شکل (c_i, x_i, σ_i) ، تابع $L.sort()$ همه عناصر L را بر اساس c_i به ترتیب صعودی مرتب می کند.



شکل ۶- توسعه برنامه ریزی مسیر RRT*.

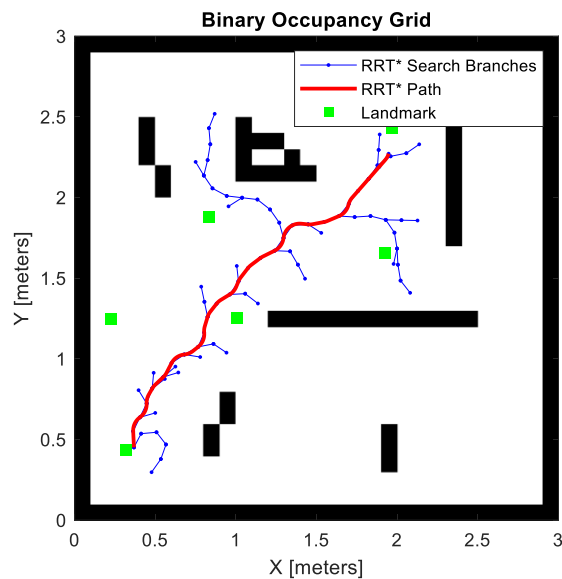
در پیکربندی، RRT* درخت جستجو را بر اساس گره ها با جمع آوری تصادفی نمونه ها به صورت تدریجی ایجاد می کند، بنابراین گودها به گونه ای ساخته می شوند که به سرعت فاصله مورد انتظار گره های انتخابی تصادفی در درخت را کاهش می یابد. همانطور که گره ها رشد می کنند، با استفاده از x_{rand} (گره های تصادفی)، برای اتصال گره x_{near} و گره x_{rand} تولید می شود [۲۶]. در شکل (۶) نشان داده شده است که با تکرار این فرآیند، گره ها با هم یک درخت را تشکیل می دهند تا به حالت هدف برسند. شبه کد RRT* بصورت الگوریتم (۱) است.

الگوریتم ۱- شبه کد RRT*.

[1] Algorithm 1: The RRT* Algorithm

- [2] 1: $V \leftarrow \{x_{init}\}; E \leftarrow \emptyset; T \leftarrow (V, E);$
- [3] 2: **for** $i = 1 \rightarrow N$ **do**
- [4] 3: $x_{rand} \leftarrow Sample(i);$
- [5] 4: $x_{near} \leftarrow Near(V, x_{rand});$
- [6] 5: **if** $x_{near} = \emptyset$ **then**
- [7] 6: $x_{near} \leftarrow Nearst(V, x_{rand});$
- [8] 7: **end if**
- [9] 8: $x_{parent} \leftarrow Find\ Best\ Parents(x_{near}, x_{rand});$
- [10] 9: **if** $x_{parent} \neq Null$ **then**
- [11] 10: $V \leftarrow V \cup \{x_{rand}\}; E \leftarrow E \cup \{(x_{near}, x_{rand})\};$
- [12] 11: $E \leftarrow Rewire(E, x_{near}, x_{rand});$
- [13] 12: **end if**
- [14] 13: **end for**
- [15] 14: **return** $T = (V, E);$

الگوریتم RRT^* مبتنی بر نمونه گیری و غیرمتمرکز است و فرآیند هدر رفتن در حضور تعداد زیادی مجموعه داده برای پردازنده را کاهش می دهد. علاوه بر این، در حین گشت زنی و کاوش، RRT^* مسیری بهینه برای دور زدن موانع پیدا می کند. این بدان معناست که ربات می تواند با استفاده از اجتناب از موانع و مسیریابی با تولید مدل های احتمالی محیط، در یک منطقه ناشناخته حرکت کند. RRT^* با در نظر گرفتن مدل ربات به واقعیت نزدیک است، بدون اینکه فرض کنیم این ربات دارای قابلیت حرکت در همه جهات است. بر اساس مدل های سینماتیکی، مشخص است که هر تعداد از موقعیت ها را می توان پس از مدت زمان مشخصی توسط ربات به دست آورد و به همین ترتیب، تعداد موقعیت جدید بر اساس موقعیت قبلی ایجاد می شود. این باعث ایجاد تعداد زیادی از مقادیر ممکن در طول زمان می شود. بر اساس هر موقعیت x_{rand} مورد انتظار، نزدیک ترین موقعیت x_{near} را همیشه می توان در تعداد زیادی از مقادیر یافت. سپس مجموعه ای از مسیرهای متناظر با معکوس کردن نقاط وضعیت به دست می آید. روش مورد استفاده در اینجا مشابه استراتژی نمونه گیری تصادفی است و سپس بهترین ورودی کنترلی از نتایج نمونه انتخاب می شود. همچنین مواجهه با مشکلات مربوط به موانع بسیار ساده است: زمانی که x_{rand} در حال برخورد با مانع است، مستقیماً نادیده گرفته می شود و اگر مسیر مانع را قطع کند، x_{near} دیگری مستقیماً انتخاب می شود. این به مجموعه ای از مسیرهای مختلف برای اجتناب از برخورد با موانع منجر می شود. سپس بر این اساس یک طرح مسیر تولید می شود و می توان نتایج را ترسیم و بررسی کرد.



شکل ۷- مسیر نامی طراحی شده توسط الگوریتم RRT^* و نقشه محیط.

۴-نتایج

ربات در محیط نقشه حرکت می کند و اطلاعات اندازه گیری را از سنسورهای داخلی حرکت چرخ های خود ثبت می کند و با حسگرهای خارجی خود اندازه گیری های مربوط به نشانه های محیط و موانع را جمع آوری می کند. محیط شبیه سازی مربوطه در متلب شبیه سازی شده است. برای حل مشکل دقت ناوبری، روش موقعیت یابی مبتنی بر EKF و روش برنامه ریزی مسیر مبتنی بر RRT^* به ترتیب معرفی شده است. محیط شبیه سازی ربات به نوعی شبیه به یک محیط بسته فرضی طراحی شده است. در شکل (۷) مسیر واقعی ربات و نقشه محیط نشان داده شده است. مسیر حرکت ربات با رنگ قرمز مشخص شده است. مسیر حرکت ربات توسط

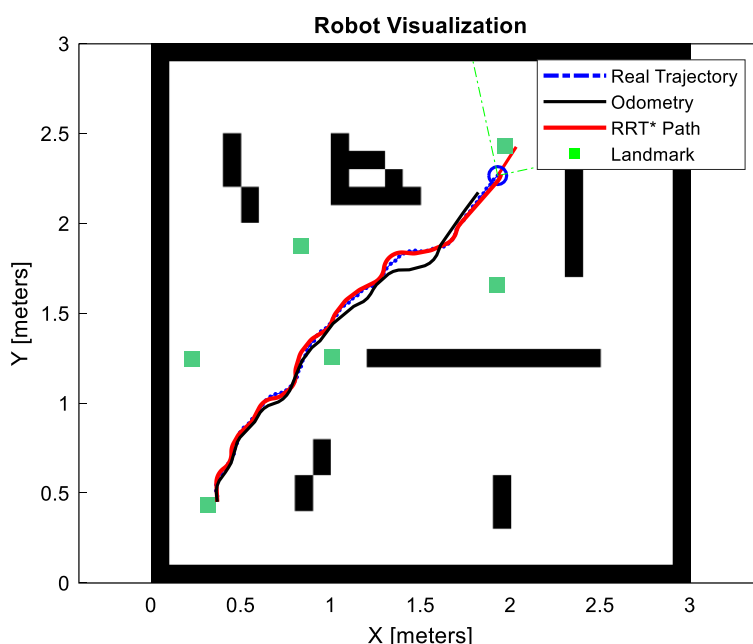
الگوریتم درختان تصادفی با کاوش سریع طراحی می شود. موقعیت یابی با شناسایی نشانه ها از داده های اندازه گیری شده و مطابق ساختن آن ها با نشانه های موجود در نقشه انجام می شود. از تفاوت بین موقعیت نشانه های تخمین زده شده و اندازه گیری شده، برای محاسبه موقعیت و وضعیت ربات استفاده می شود. این ربات مجهز به حسگر لیدار است. حسگر لیدار در جلوی ربات نصب شده است و فاصله و زاویه نسبت به نشانه مشاهده شده را اندازه گیری می کند. سرعت ربات 1 m/s است و ماکزیمم زاویه فرمان آن $1/59$ درجه در نظر گرفته شده است. فاصله بین چرخ ها $0/05$ و ماکزیمم دامنه فاصله یاب لیزری $0/5$ متر و میدان دید آن 90 درجه است. فرکانس کنترل 50 Hz و انحراف استاندارد نویز سرعت مستقیم و سرعت زاویه ای به ترتیب σ_v و σ_γ بصورت زیر است:

$$\sigma_v = 0.01 \text{ m/s}, \sigma_\gamma = 0.017^\circ$$

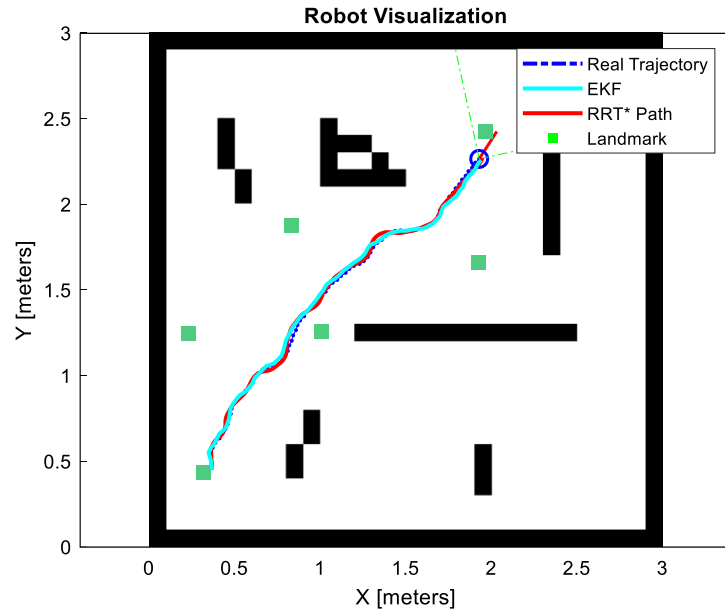
فرکانس نمونه برداری سنسورها 5 Hz و انحراف استاندارد نویز اندازه گیری در فاصله و سمت به ترتیب σ_r و σ_θ بصورت زیر است:

$$\sigma_r = 0.5 \text{ m/s}, \sigma_\theta = 0.05^\circ$$

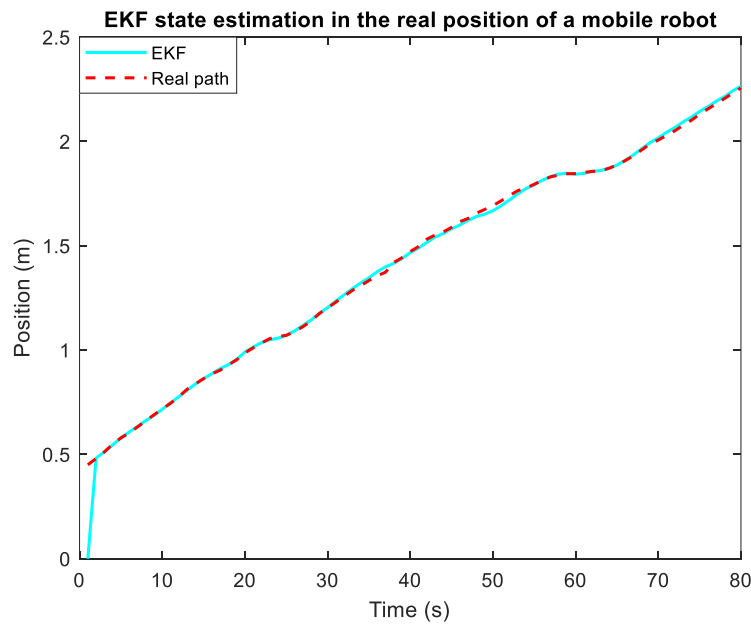
شکل های (۸) و (۹)، نتایج حاصل از موقعیت یابی بر اساس ادومتری و فیلتر کالمن توسعه یافته را نشان می دهند. نشانه های موجود در نقشه با علامت ■ مشخص شده است. در شکل (۹) خط مشکی مربوط به موقعیت یابی ربات توسط ادومتری است. در شکل (۹) رنگ فیروزه ای مربوط به تخمین EKF است. کاملاً مشخص است که نتایج عملکرد روش فیلتر کالمن توسعه یافته بهتر از ادومتری است. به عبارت دیگر، در موقعیت یابی با الگوریتم پیشنهادی، مسیر تخمین زده شده وسیله تا حد امکان نزدیک به مسیر واقعی می باشد. ربات از نقطه شروع تا نقطه هدف را با موفقیت موقعیت یابی کرده و طبق برنامه ریزی مسیر که توسط الگوریتم RRT^* طراحی شده، پیش رفته و به نقطه هدف رسیده است. در ترکیب الگوریتم EKF و RRT^* مشخص است که مسیر تولید شده توسط شبیه سازی الگوریتم پیشنهادی بهینه است. در شکل (۱۰) دقت موقعیت یابی EKF آمده است.



شکل ۸- مسیر واقعی و تخمین موقعیت ربات توسط ادومتری.

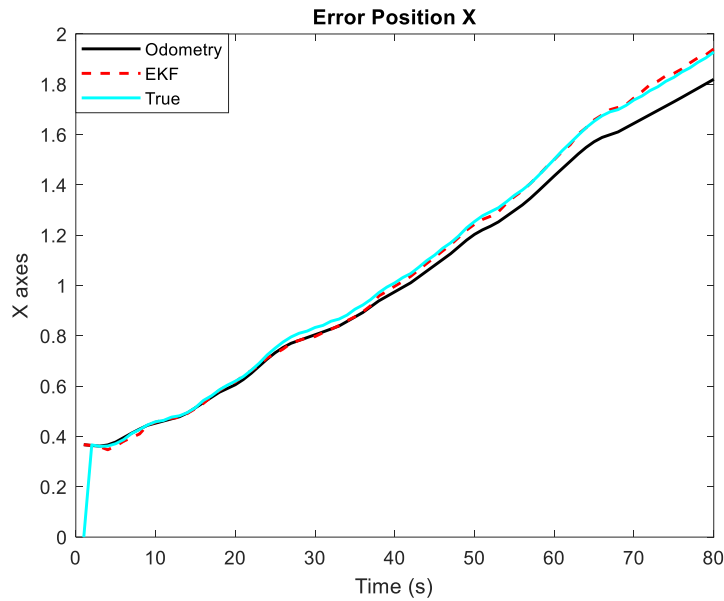


شکل ۹- مسیر واقعی و تخمین موقعیت ربات توسط EKF.



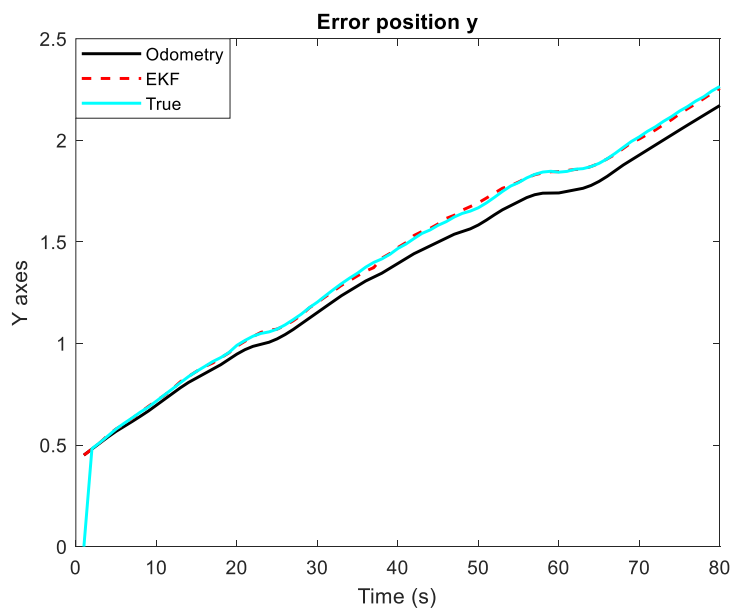
شکل ۱۰- دقت تخمین EKF در رابطه با تخمین موقعیت حقیقی ربات متحرک.

در شکل (۹)، خطای موقعیت یابی و خطای زاویه ایجاد شده بسیار کوچک است و خطا افزایشی نیست. در نتیجه، رویکرد پیشنهادی موجب افزایش دقت موقعیت یابی به طور قابل توجهی در مقایسه با ادمتری شده است. در شکل های (۱۱) تا (۱۳) خطای موقعیت یابی در محورهای X و Y و همچنین جهت گیری زاویه جلوی ربات نسبت به زاویه مسیر آمده است.

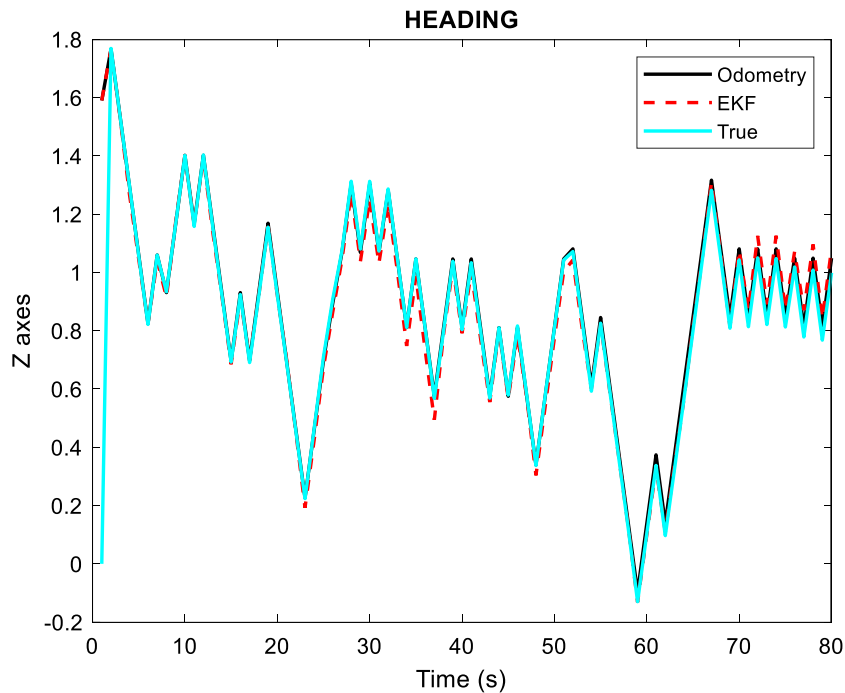


شکل ۱۱- دقت تخمین EKF در مقایسه با محاسبات ادومتری در محور X .

در شکل (۱۱) خطای موقعیت یابی EKF با ادومتری در محور X مقایسه شده است ادومتری تنها در همان لحظات ابتدایی مطابق موقعیت واقعی ربات عمل کرده و انباشته شدن نویزها باعث عملکرد ناموفق ادومتری شده است این به دلیل در نظر نگرفتن نویز سنسورها و نویزهای مربوط به سرعت زاویه ای و سرعت پیشروی است. اما، در خصوص موقعیت یابی با EKF به دلیل اینکه مشخصات نویزها در نظر گرفته شده اند و اطلاعات حسگرهای درونی و بیرونی با هم ترکیب شدند، دیگر ربات با مشکل انباشته شدن خطاها مواجه نمی شود. در شکل (۱۲) خطای موقعیت یابی EKF و ادومتری در محور Y مقایسه شده است. خطاهای موقعیت یابی مانند خطای سیستم و خطای حسگرها که با استفاده از الگوریتم ادومتری ایجاد می شود، به طور پیوسته در حال جمع شدن هستند.



شکل ۱۲- دقت تخمین EKF در مقایسه با محاسبات ادومتری در محور Y .



شکل ۱۳- دقت تخمین EKF در مقایسه با محاسبات ادومتری مربوط به زاویه دیفرانسیل ربات در جهت گیری.

در شکل (۱۳) جهت گیری زاویه جلوی ربات تقریباً به درستی زاویه مسیر را دنبال کرده است. جذر مربع خطا^۱ (RMSE) معیاری است که معمولاً برای اندازه گیری تفاوت بین مقادیر پیش بینی شده و مقادیر واقعی در تحلیل رگرسیون استفاده می شود. RMSE میانگین انحراف یا خطا بین مقادیر پیش بینی شده و واقعی پیش بینی های یک مدل را نشان می دهد. با گرفتن جذر میانگین مجذور خطاهای بین مقادیر پیش بینی شده و مقادیر واقعی محاسبه می شود. مقدار RMSE کمتر نشان دهنده برازش بهتر مدل است. به منظور ارزیابی جامع تر کارایی روش پیشنهادی، عملکرد آن با دو روش مرجع مبتنی بر فیلتر کالمن توسعه یافته و یک روش مبتنی بر ادومتری، مطابق با مطالعات اخیر مقایسه شده است. نتایج این مقایسه در جدول X ارائه شده است. همان گونه که مشاهده می شود، روش پیشنهادی، در تمامی شاخص های کلیدی شامل میانگین خطای جهت X و Y، خطای زاویه دیفرانسیل و مقدار RMSE، عملکرد بهتری نسبت به سایر روش ها از خود نشان داده است. همچنین زمان محاسباتی این روش قابل رقابت با روش های رقیب است. این نتایج نشان دهنده دقت بالاتر و پایداری بیشتر روش پیشنهادی در ردیابی و ناوبری ربات متحرک است. به طور خاص، مقدار RMSE به دست آمده برای روش پیشنهادی ۰/۰۵۵ متر است که نسبت به روش های به ترتیب بهبود قابل توجهی را نشان می دهد. همچنین خطای زاویه و میانگین خطا در هر دو راستا نیز در روش پیشنهادی کمتر از سایر روش ها است و این موضوع بیانگر افزایش دقت تخمین موقعیت و جهت یابی ربات است. در مجموع، نتایج جدول X تأیید می کند که ترکیب فیلتر کالمن توسعه یافته با الگوریتم پیشنهادی می تواند منجر به بهبود معنی دار عملکرد ردیابی و ناوبری در سناریوهای عملیاتی شود.

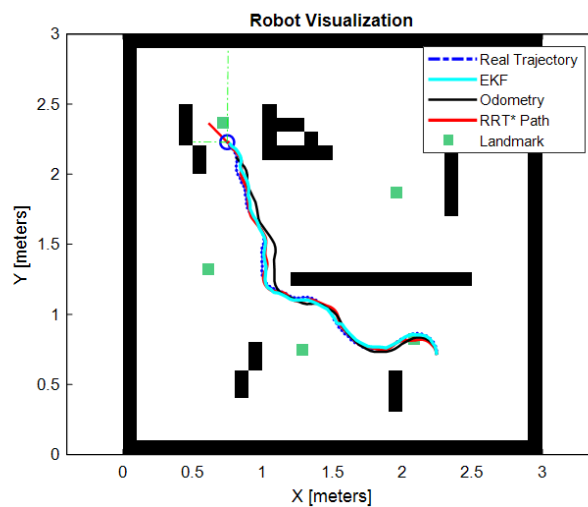
¹ Root Mean Square Error

جدول ۱- مقایسه عملکرد روش پیشنهادی با سایر روشها

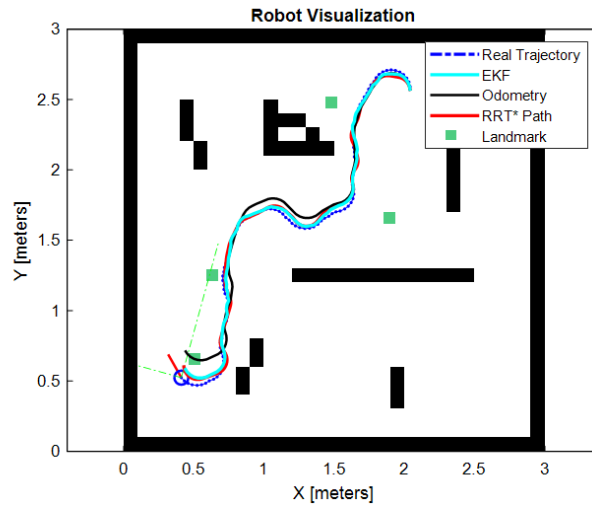
استراتژی ناوبری [۲۰]	[۱۹] میانگین خطا (متر)		[۱۸] خطای زاویه دیفرانسیل (درجه)	RMSE [۱۷]	[۱۶] زمان محاسبات (ثانیه)
	[۲۲] درجهت X	[۲۱] ذر جهت Y			
EKF + RRT* [۲۸]	۰/۰۲۷ [۲۷]	۰/۰۲۵ [۲۶]	۰/۰۲۵ [۲۵]	۰/۰۵۵ [۲۴]	۱/۵۷ [۲۳]
IEKF [۲۸]	۰/۰۸ [۳۳]	۰/۰۷ [۳۲]	۰/۰۴ [۳۱]	۰/۰۶۹ [۳۰]	۱/۷ [۲۹]
EKF [۲۷]	۰/۰۹۹ [۳۹]	۰/۰۹ [۳۸]	۰/۰۴۵ [۳۷]	۰/۰۷۴ [۳۶]	۱/۵ [۳۵]
Odometry [۴۶]	۰/۱۱۵ [۴۵]	۰/۱۰۲ [۴۴]	۰/۰۵۵ [۴۳]	۰/۰۸۵ [۴۲]	۱/۳۴ [۴۱]

۴-۱- بررسی عملکرد در مسیرهای مختلف

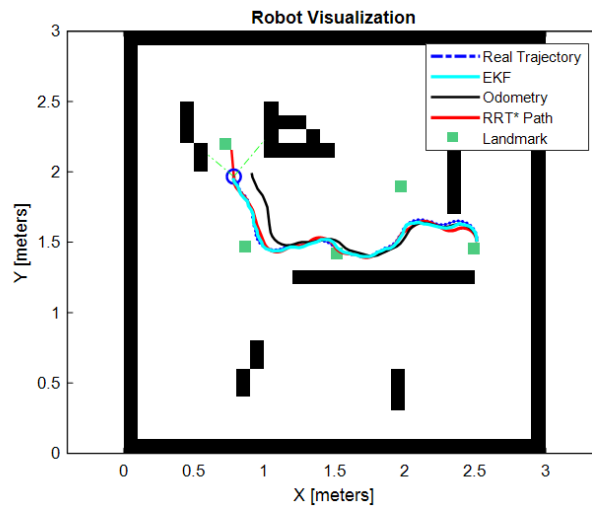
برای بررسی بهتر الگوریتم برنامه ریزی مسیر، عملکرد RRT* در حالت های مختلف آزمایش شد. شبیه سازی در چهار گروه مختلف از مسیرهای انتخابی انجام می شود. ربات متحرک از نقطه شروع تا نقطه هدف در پایین حرکت می کند. مسیرهای قرمز نشان دهنده نتایج الگوریتم RRT* است. مسیرهای سیاه نشان دهنده نتایج بر اساس محاسبات اودومتري هستند. مسیرهای فیروزه ای نشان دهنده تخمین موقعیت توسط الگوریتم EKF است و نقاط سبز رنگ نشانه های نقشه هستند. مسیرهای قرمز نشان دهنده نتایج الگوریتم RRT* است. شکل های (۱۴) تا (۱۷) نتایج شبیه سازی را نشان می دهند.



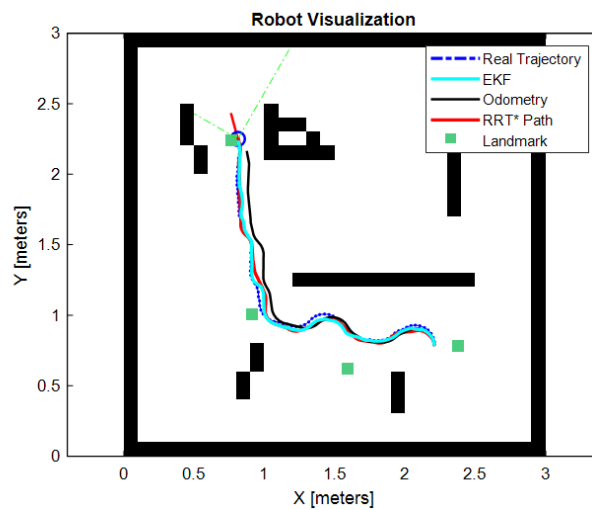
شکل ۱۴- مسیر اول تخمین زده شده توسط الگوریتم EKF.



شکل ۱۵- مسیر دوم تخمین زده شده توسط الگوریتم EKF.



شکل ۱۶- مسیر سوم تخمین زده شده توسط الگوریتم EKF.



شکل ۱۷- مسیر چهارم تخمین زده شده توسط الگوریتم EKF.

بر اساس نتایج بخش (۴)، به وضوح مشخص است که الگوریتم مسیریابی RRT^* به طور موثر مسیر بهینه را می سازد. نتایج نشان می دهد که الگوریتم RRT^* مسیر را به گونه ای طراحی می کند که الزامات ایمنی رعایت می شود و محدودیت های تعریف شده توسط محیط عملیاتی، مورد توجه ویژه واقع می شود.

۵- نتیجه گیری

در این مقاله، موقعیت یابی ربات متحرک در محیط های داخلی با استفاده از ترکیب فیلتر کالمن توسعه یافته و الگوریتم مسیریابی درخت تصادفی ارائه شده است. در روش پیشنهادی، مسئله موقعیت یابی تبدیل به یک مسئله بهینه سازی شده و سپس از RRT^* برای برنامه ریزی مسیر ربات استفاده شده است. عملکرد موقعیت یابی مبتنی بر EKF با عملکرد موقعیت یابی با ادومتری از نظر دقت و درصد خطا مورد ارزیابی قرار گرفته است. نتایج شبیه سازی ها نشان می دهند که عملکرد موقعیت یابی بر اساس الگوریتم EKF بهتر از موقعیت یابی بر اساس ادومتری است. یکی از مزایای موقعیت یابی بر اساس الگوریتم EKF در این است که این الگوریتم مانند موقعیت یابی بر اساس ادومتری تنها وابسته به اطلاعات حسگرهای درونی نیست. بنابراین خطای موقعیت یابی کاهش می یابد. سپس، روش RRT^* برای یافتن یک مسیر امکان پذیر با توجه به محیط عملیاتی معرفی می شود. روش RRT^* اهداف برنامه ریزی مسیر را برآورده می کند که این اهداف عبارتند از: (۱) مسیر بهینه در سریع ترین زمان ممکن ایجاد شود. (۲) الزامات ایمنی کاملاً رعایت شوند. (۳) محدودیت های تعریف شده توسط محیط عملیاتی، مورد توجه ویژه واقع شوند. نتایج نشان می دهند که ربات متحرک می تواند مسیر ایده آلی را در محیط های داخلی باریک پیدا کند که کارایی ناوبری مستقل و کیفیت مسیر را تضمین می کند. خطای موقعیت یابی و خطای زاویه ایجاد شده توسط روش پیشنهادی بر اساس الگوریتم EKF بسیار کمتر از الگوریتم ادومتری است و خطا با گذشت زمان افزایش نخواهد یافت. مربعات خطای موقعیت یابی $0/055$ است. در مقایسه با الگوریتم ادومتری، میانگین دقت موقعیت یابی روش پیشنهادی 73% افزایش یافته است. بهبود عملکرد از طریق شبیه سازی های متعدد تأیید می شود.

مراجع

- [1] Siegwart, R., Nourbakhsh, I.R., and Scaramuzza, D.: 'Introduction to autonomous mobile robots' MIT press, 2011.
- [2] Zheng, L., Zhan, X., Zhang, X., Wang, S., and Yuan, W.: 'Heading estimation for multimode pedestrian dead reckoning', *IEEE Sensors Journal*, vol.20, no.15, pp. 8731-8739, 2020.
- [3] Madray, I., Suire, J., Desforges, J., and Madani, M.R. 'Relative angle correction for distance estimation using K-nearest neighbors', *IEEE Sensors Journal*, vol. 20, no. 14, pp. 8155-8163, Jul. 2020
- [4] Guo, S., Zhang, Y., Gui, X., and Han, L.: 'An improved PDR/UWB integrated system for indoor navigation applications', *IEEE Sensors Journal*, vol. 20, no. 14, pp. 8046-8061, Jul. 2020.
- [5] Rauniyar, S., Bhalla, S., Choi, D., and Kim, D.: 'EKF-SLAM for Quadcopter Using Differential Flatness-Based LQR Control', *Electronics*, vol. 12, no. 5, p. 1113, 2023.
- [6] Leonard, J.J., and Durrant-Whyte, H.F.: 'Directed sonar sensing for mobile robot navigation', Springer Science & Business Media, 2012.
- [7] Chen, L., Hu, H., and McDonald-Maier, K.: 'EKF based mobile robot localization', in Proc. IEEE Int. Conf. on Industrial Technology (ICIT), Athens, Greece, 2012, pp. 149-154.

- [8] Suliman, C., Cruceru, C., and Moldoveanu, F.: 'Mobile robot position estimation using the Kalman filter', *Acta Marisiensis, Seria Technologica*, vol. 6, pp. 75–80, 2009.
- [9] Zhang, H., Chen, N., and Fan, G.: 'An Improved Localization Algorithm for Intelligent Robot', in *Proc. IEEE Int. Conf. on Artificial Intelligence and Big Data (ICAIBD)*, Chengdu, China, 2019, pp. 1–5.
- [10] Lin, H.-Y. and M.-C. Yeh, 'Drift-Free Visual SLAM for Mobile Robot Localization by Integrating UWB Technology'. *IEEE Access*, vol. 10, pp. 93636–93645, 2022.
- [11] Coelho, F.O., et al. 'EKF and computer vision for mobile robot localization', in *Proc. 13th APCA Int. Conf. Automatic Control and Soft Computing (CONTROLO)*, 2018.
- [12] Zhang, F., et al. 'A novel strategy of localization based on EKF for mobile robot', in *Proceedings of the 33rd Chinese Control Conference*. 2014.
- [13] Chen, L., H. Hu, and K. McDonald-Maier. 'EKF based mobile robot localization', in *Proc. 3rd Int. Conf. Emerging Security Technologies*, 2012.
- [14] Joon, A. and W. Kowalczyk. 'Leader Following Control of Non-holonomic Mobile Robots Using EKF-based Localization' in *2023 27th International Conference on Methods and Models in Automation and Robotics (MMAR)*. 2023.
- [15] Sun, Z., et al., 'Multi-Risk-RRT: An Efficient Motion Planning Algorithm for Robotic Autonomous Luggage Trolley Collection at Airports', *IEEE Transactions on Intelligent Vehicles*, vol. 9, pp. 3450-3463, 2024.
- [16] Mizuno, M. and T. Kubota. 'A new path planning architecture to consider motion uncertainty in natural environment', in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. 2020. IEEE.
- [17] Cao, M., Zhou, X., and Ju, Y., 'Robot Motion Planning Based on Improved RRT Algorithm and RBF Neural Network Sliding', *IEEE Access*, vol. 11, pp.121295-121305, 2023.
- [18] Yu, Z., Jiang, X., and Liu, Y., 'Pose estimation of an aerial construction robot based on motion and dynamic constraints', *Robotics and Autonomous Systems*, vol. 172, p. 104591, 2024.
- [19] Brigadnov, I., Lutonin, A., and Bogdanova, K., 'Error State Extended Kalman Filter Localization for Underground Mining Environments', *Symmetry*, vol. 15, no. 2, p. 344, 2023.
- [20] Panigrahi, P.K., and Bisoy, S.K.: 'Localization strategies for autonomous mobile robots: A review', *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 8, pp. 6019-6039, 2022.
- [21] Kong, F., Chen, Y., Xie, J., Zhang, G., and Zhou, Z.: 'Mobile robot localization based on extended kalman filter', in *Proceedings of IEEE Conference*, 2006, pp. 9242–9246 .
- [22] Karaman, S. and E. Frazzoli. 'Optimal kinodynamic motion planning using incremental sampling-based methods', in *49th IEEE conference on decision and control (CDC)*. 2010.
- [23] Karaman, S. and Frazzoli, E. 'Sampling-based algorithms for optimal motion planning', *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.
- [24] Noreen, I., Khan, A., and Habib, Z., 'Optimal path planning using RRT* based approaches: a survey and future directions', *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 11, 2016.

-
- [25] Szabat, K., et al., 'A fuzzy unscented Kalman filter in the adaptive control system of a drive system with a flexible joint', *Energies*, vol. 13, no. 8, p. 2056, 2020.
- [26] Woo, R., E.-J. Yang, and D.-W. Seo, 'A fuzzy-innovation-based adaptive Kalman filter for enhanced vehicle positioning in dense urban environments', *Sensors*, vol. 19, no. 5, p. 1142, 2019
- [27] C.-Y. Yang, H. Samani, Z. Tang, and C. Li, 'Implementation of extended Kalman filter for localization of ambulance robot', *International Journal of Intelligent Robotics and Applications*, vol. 8, no. 4, pp. 960–973, Jun. 2024.
- [28] Y. Wang, Q. Zhang, and J. Liu, 'Application of a Modified IEKF Algorithm in Mobile Robot Localization', in *Proc. 2022 4th Int. Conf. on Robotics, Intelligent Control and Artificial Intelligence (RICAI)*, Guangzhou, China, Dec. 2022, pp. 53–57.